DOCUMENT RESUME

ED 066 796 EA 004 405

AUTHOR	Wilson, Stanley; Billingsley, Ray
TITLE	FACTOR - FACTOR II. Departmental Program and Model Documentation 71-3.
INSTITUTION	Texas A and M Univ., College Station.
PUB DATE	Jan 71
NOTE	75p.; Presentation prepared in cooperation with the State NUPLEX Program: Nuclear-Agro-Industrial-Complex Study
AVAILABLE FROM	Dept. of Agricultural Economics & Rural Sociology, Texas A & M University, College Station, Texas 77843 (Free)
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Computer Assisted Instruction; *Computer Oriented Programs; Computer Programs; *Cost Effectiveness; *Economic Education; *Economics; *Educational Administration; Management; Operations Research; Productivity
IDENTIFIERS	Factor; Factor II

ABSTRACT

This computer program is designed to optimize a Cobb-Douglas type of production function. The user of this program may choose isoquants and/or the expansion path for a Cobb-Douglas type of production function with up to nine resources. An expansion path is the combination of quantities of each resource that minimizes the cost at each production level. At each production level, the program will also output the total cost, total fixed cost, total variable cost, average variable cost, average fixed cost, and average and marginal cost. In addition, it will calculate the marginal product and the ratio of the value of the marginal product to the price of the resource for each resource. Finally, it will output the marginal rate of technical substitution and resource price ratios between each resource and every other resource. At the option of the user, the profit maximizing level of output will also be sought. The user may also choose to hold one or more resources fixed in quantity. Changes in the function's parameters or prices may be made and the output regenerated up to 30 times in the same run. (Author)

ED 066796

Departmental Program and Model Documentation

FACTOR - FACTOR II

EA 004 405

71-3

Texas A&M University
The Texas Agricultural Experiment Station
H. O. Kunkel, Acting Director

ED 066796

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

FACTOR - FACTOR II

Departmental Program and Model Documentation 71-3

By

Stanley Wilson and Ray Billingsley

Department of Agricultural Economics and Rural Sociology
Texas Agricultural Experiment Station
College Station, Texas 77843

in cooperation with the

State NUPLEX Program:
Nuclear-Agro-Industrial-Complex Study
January 1971

CONTENTS OF PROGRAMMING AND MODEL DOCUMENTATION

I. PROGRAM IDENTIFICATION AND BACKGROUND

1. PROGRAM NAME . . . . . FACTOR FACTOR II: Optimizes a Cobb-Douglass Production Function
2. DOCUMENTATION NUMBER . . . 71-3
3. AUTHOR OF DOCUMENTATION. . Stanley Wilson and Ray Billingsley
4. PROGRAMMER . . . . . Stanley Wilson
5. ORIGIN OF PROGRAM . . . . James Wherley [1]
6. LANGUAGE/COMPUTER. . . . . Fortran IV / IBM 360/65
7. DATE . . . . . November 1, 1970

II. USER DOCUMENTATION

	Page Number *
1. GENERAL ABSTRACT	2
2. OPERATIONAL USE	3
Data Requirements/Options.	3
Input Specification/Format	4-12
Jobstream.	13

III. PROGRAMMER DOCUMENTATION

1. GENERAL DESCRIPTION	14
Purpose	14
Assumptions and Capabilities	14
Algorithm.	15-18
Mathematical Description	18-20
References	20
2. PROGRAM DESCRIPTION (Main and Subprograms)	49
Program Abstract	20
Variable List.	20
Subroutines/Functions called	21-22
Flowchart.	23-25
3. PROGRAM LISTING	48
4. SAMPLE PROGRAM RUN	63
Input File/Sample Data	64
Output File	66

* If a particular element is not contained in this documentation place XXX in this column.

II. USER DOCUMENTATION

1. GENERAL ABSTRACT

FACTOR-FACTOR II (F-F II) is a more general version of F-F-I.[2] By exercising certain options written into F-F II, all functions of F-F-I, can be performed and in addition, one or more expansion paths can be generated. An expansion path is the combination of quantities of various resources which will minimize cost at each specified level of output. There are two types of expansion paths, the general and the partial. In the partial, one or more resources are fixed in quantity and the rest are allowed to go to that level which minimizes cost for a specified level of output. In the general, all resources are allowed to go to that level which minimizes cost at each specified level of output. The term expansion path will be reserved for the general expansion path. Unless told otherwise, the program will generate a general expansion path. The program will also generate a table showing the ratio between the value of the marginal product and the cost (price) of each resource. The profit maximizing level of output is obtained when this ratio equals 1.0. If the user wishes, the program will change the output level by smaller and smaller amounts as the profit maximizing level of output is approached so that the program will converge on the exact profit maximizing quantity. If not, the program will simply increase output each time by a specified amount.

2. OPERATIONAL USE

Data Requirements/Options: This program is based on the Cobb-Douglass production function which is of the form: $Y = AX_1^{B_1} X_2^{B_2} X_3^{B_3} \ldots X_n^{B_n}$. This program is limited to nine resource variables ($1 < n < 10$).

Y = Quantity of the Product

X_n = Quantity of the n'th resource

The coefficients of the function are A and B_1, B_2, B_3,...B_n. Since output from the program includes isoquants, expansion paths or both, at the option of the user, the required input data depends on which type of output the user wants.

Data Requirements and Isoquant Only: The program changes the quantity of one resource, called the independent resource, or X(INDE), and calculates how much of a second resource, called the dependent resource or X(LDE), is needed, holding all other X values and Y constant. This is repeated with X(INDE) changed by CHX until XLAST is reached. This is an isoquant. Then Y is changed by CHY and the isoquant is regenerated again and again until Y reaches YLAST. The program requires values for A, all X's except X(LDE), CHX, XLAST, CHY, YLAST, and the B's. The user has a choice as to which he will specify, a beginning level of Y or a beginning dependent resource quantity. If the user specifies a beginning level of Y, then the beginning quantity of the dependent resource should be zero (that is, the space on the data card should be blank). If the beginning quantity of the

dependent resource is specified, the original value of Y should be zero (that is, the space on the data card should be blank). Then the program will calculate the first quantity of Y using the quantities of all resources.

An isoquant will be generated at level Y for various quantities of the independent resource changing the dependent resource as required. When XLAST is reached, CHY will be added to the calculated value of Y and a second isoquant generated. This continues until the entire field is generated.

Data Requirements for Expansion Path Only: The program requires a value for"A" for each of the resource coefficients (B's), for each of the resource prices (PX's), a beginning level of Y, a final level of Y, (YLAST), a value for X(INDE) and an amount that Y will change by (CHY). The user may also elect to give a value for the price of the output, PY. This will make the cost and revenue value table more meaningful but the program will run if PY is zero (left blank). The same applies to fixed cost, FC.

Input Specifications/Format: There are a number of options in the program and these are usually associated with and controlled by intergers in the input data, through the numeric values given these intergers. The input or data deck is divided into three sections. In order of input they are (i) the description cards, (ii) original set of parameter values, and (iii) the changes in parameter values.

(i) The description cards are used to print out a brief description of the problem and to print a one line identification at the top of each page of output. The user may have as many description cards as he wishes. The description may be entered in any column from 5 to 80 inclusive. The last card must have a number in Column 1 or 2. This number signals the program that this is the last description card.

The contents of all description cards will be printed out on a separate page. The contents may be alphanumeric and or special symbols. The contents of the last card will be printed at the top of each page of output and can be used as a brief identification of the problem.

(ii) The cards which give the initial set of values for the parameters of the model follow the description cards.

In order of Input they are:

[1] The Y card Input

[2] The A card Input

[3] Resource Cards

[1] The Y, Card Input	CARD COLUMN	FORMAT
IFLAG If IFLAG = 0, Y changes by CHY If IFLAG = 1, Y changes to converge on the profit maximizing level of output. (Where $(VMP_j / P_j) = 1.0$)	1-2	I 2
Y First value for Y	5-24	F20.4
CHY Change in Y If CHY $\leq$ 1., Y is increased by a percentage amount, CHY% If CHY > 1., Y is increased by the amount of CHY	25-44	F20.4
YLAST The last value Y will reach before changes in the parameters are introduced	45-64	F20.4
PY The price per unit of Y	65-78	F14.3
LX Determines which tables will be printed 01 - prints both isoquants and expansion path tables 11 - prints expansion path only 22 - prints isoquants only	79-80	I 2

Explanation: Y is the original level of output or value of Y. CHY is the change in Y and YLAST is the highest value calculated for Y. If CHY is greater than 1.0 then CHY is added to Y to produce each new isoquant. If CHY is less than 1.0 then Y is increased by CHY% to produce each new isoquant. Thus if CHY is 0.05, Y will increase by 5% each time. In this way the user may choose an absolute amount of growth or a growth rate.

[2] The A, Card Input

	CARD COLUMN	FORMAT
IFLAG Indicates the manner in which X and A are changed in calculating the isoquant and expansion paths. b - increases X(I) that is specified as X(INDE) if CHX $\leq$ 1., change is CHX% if CHX$>$1., change by absolute amount, CHX 2 - increase A by CHX (absolute amount) X(INDE) not changed 3 - increase A by CHX%, X(INDE) not increased.	1-2	I 2
A The Y intercept	5-24	F20.4
CHX Change in specified X	25-44	F20.4
XLAST Last value for specified X	45-64	F20.4
FC - fixed cost	65-78	F14.3
IA Indicates the form in which A is entered. The value of A may be entered as A, log 10 A or log e A b - Log e A entered and used in calculations 10 - Real A is entered and used in calculations 20 - Log 10 A entered and used in calculations	79-80	I 2

Explanation: A may be entered as its own value, as the log to the base 10 or as the log to the base e of A. IA is used to indicate which of the three values are entered. If IA is 00 then the value is taken to be the log to the base e of A. If IA is 10, then the value is taken to be the value of A. If IA is 20, then the value is taken to be the log to the base 10 of A. CHX is the change in the independent resource. XLAST is the last value the independent resource will take before a new isoquant is begun (before Y is changed).

[3] Resource Cards	CARD COLUMN	FORMAT
IFLAG Controls whether X(I) will be held constant or set to its cost minimizing quantity by the program. If all IFLAG's on all resource cards are blank then the program produces a general expansion path. For all resource cards except the last card: b - X(I) will be set to its cost minimizing quantity 1 - X(I) will be held constant For the last resource card: 2 - X(I) will be set to its cost minimizing quantity 3 - X(I) will be held constant	1-2	I 2
X(I) Enter the value for X if it is to be specified. If X(I) is to be calculated, this field should be left blank and a 1 entered in CC 80. If X(I) is changed by CHX enter a 2 in CC 80.	5-24	F20.4
B(I) The regression coefficient	25-49	F25.12
EX(I) - not used	50-64	F15.9
PX(I) - price of the I'th resource	65-78	F14.3
IX(I) Determines the X which is specified and the one to be calculated 1 - indicates that X(I) is calculated 2 - indicates that X(I) is specified and will be increased by CHX b - indicates that X(I) is held constant	79-80	I 2

Explanation: The program will handle up to nine different resources and there must be one resource card for each different resource. Since there are a variable number of resource cards IFLAG is used to tell the program which is the last resource card. The resource identification numbers (the value of I) are assigned in sequence. The first card is taken to be the value for resource 1, and X(I) is X(1) as I = 1. On the second card I = 2 and so forth for up to nine cards. B(I) is the coefficient of the X(I) resource (see Mathematical Description). IX(I) is used to tell the

program which of the several resources is the dependent and which is the independent resource. If IX(I) = 1 (if there is a 1, in Column 80), then the I'th resource is taken to be the dependent resource. If XI(I) = 2 (if there is a 2, in Column 80), then the I'th resource is taken to be the independent resource. IX(I) should be blank on all other resource cards.

(iii) After the cards which set the original set of parameters comes the change cards. There may be more than one set of change cards. Each set introduces changes into the previous values of the parameters and generates a complete set of isoquants, (that is, an isoquant field and/or a new expansion path). With two sets of change cards, the program will generate three isoquant fields, and/or three expansion paths, one for the original set of parameters and one for each of the two sets of change cards. In general, a user will want to make changes in only one or a few parameters in each set. While it is possible in a given change set to have a change card for each card in the initial values set (that is a change in Y card values, a change in A card values and 2 to 9 changes in resource card values), this is not likely. Thus the program must be told which card in the initial value set a given change card refers to, since there is likely to be only two or three change cards. This is accomplished by using the first two columns of the change card (Symbol = IFLAG). If IFLAG = 20 (if there is a 2 in Column 1), the change card is taken to be changed in the value of the Y card. If IFLAG = 10 (if there is a 1 in Column 1), the change card is taken to be changed in the values of the A card. If IFLAG = 00 (if Columns 1 and 2 are blank), then the change card is taken to be changed in the values on a resource card.

In the last situation (IFLAG = 00), the program must also be told which of the several resource cards the change card refers to. This is accomplished by using the last two columns of the change card (Symbol = JX), which will be taken to be the number of the resource card being changed. If JX = 4 (Column 80 contains a 4), then the change card is taken to be changed in the values of X(4), that is, changes in values on the fourth resource card.

CHANGE CARD SYMBOLS

Y Card	A Card	Resource Cards	CARD COLUMN	FORMAT
IFLAG = 20	IFLAG = 10	IFLAG = 00	1 - 2	I 2
DELY (J)	DELA (J)	DELX (I,J)	5 - 24	F20.4
DELCHY (J)	DELCHX (J)	DELB (I,J)	25 - 49	F25.12
DELYLAST (J)	DELXLAST (J)	DELEX (I,J)	50 - 64	F15.9
DELPY (J)	DELPA (J)	DELPX (I,J)	65 - 78	F14.3
IDY (J)	IDA (J)	JX (J)	79 - 80	I 2

Where DEL = Change in I = JX(J). J = number of change card sets from J = 1, 30. The end of each set of change cards is indicated by a card with a 2 in Column 2. The end of the last change card set is indicated by a card with a 6, in Column 2. After the last card in the last change set, the user may wish to terminate the problem. He can do this by putting a card with a 99 in Columns 1 and 2 after the last change card in the last change set. If instead the user uses a card with a 6 in CC1, he may enter a data set for a second problem beginning with one or more description cards. He may have any number of data sets and problems but they must each be separated by a card with a 6 in Column 2. After the last problem set must come a card with a 99 in CC1-2.

I = JX(I). JX is the number of the resource being changed. Thus if JX(J) = 4, then the changes on the card with IFLAG blank or = 02 refer to

the changes in X4. Changes may be introduced in any order.

If IDA(J)>0, the number of change periods = IDA(J). If IDA(J)>1, A grows by DELA percent for IDA(J) periods and B(I) grows by DELB(I,J) percent for IDA periods. In this way the user may choose to generate a growth rate in one or more coefficients for up to 30 periods and he can do this with one or two change cards.

OPTIONS: If IDA(J) is:

b - A and B(I) increase by absolute amounts, DELA and DELB for one period.
01 - A and B(I) increase by DELA% and DELB% respectively for one period.
02-99 - A and B(I) increase by DELA% and DELB% respectively for IDA periods.

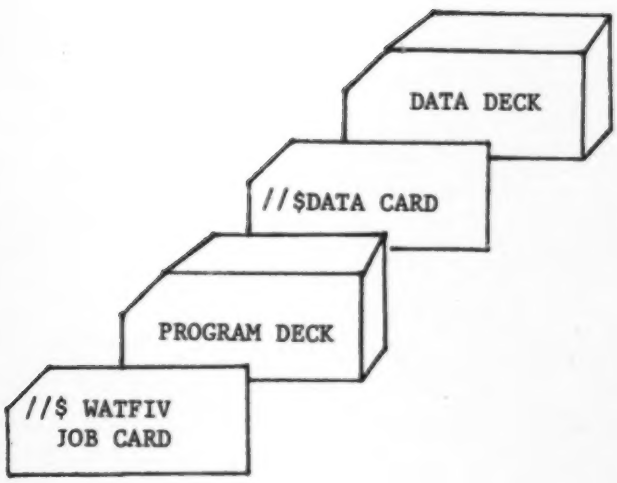
IFLAG OPTIONS:

10 - reads the values on card as changes in A, CHX, XLAST, PA
20 - reads the values on card as changes in Y, CHY, YLAST, PY
bb - indicates a resource card

The user can create several changes in the isoquant field and/or expansion path by using only one change set. He can do this by entering a number in the space for IDA(1). The number of change sets and thus isoquant fields and/or expansion paths will be equal to the value of IDA(1). At the same time, raising the value of IDA(1) above "00" causes DELA(J) to be interpreted as a "%" change in "A" and DELB(I,J) as a "%" change in B(I) instead of as absolute changes in "A" and B(I). Taken together these features allow a given growth in productivity to be simulated over a number of periods using one or two cards.

This feature can be used to simulate technological change. For example, if "A" grows at a given "%" while B(I) and "Y" stay constant, then the input quantities per unit of output will be smaller and this can be viewed as a neutral technological change. This is because since the values of B(I) are constant there is no substitution between the resources. All are saved at the same rate. But if some B(I)'s change while others do not, then these resources will be saved and the technological change will be non-neutral. By changing various parameters, each change period and not changing others and by having "Y" and/or "A" change by "%" instead of absolute amounts it is possible to simulate any type of technological change, either a neutral one or one which saves one or more resources but does not save other resources.

Jobstream



III. PROGRAMMER DOCUMENTATION

1. General Description

Purpose: F-F II serves several basic purposes. One is to generate isoquant fields, a second is to generate expansion paths and related physical input-output combinations given the price of resources and the coefficients of production. A third is to find the profit maximizing quantity of output by using the ratio of the value of the marginal product and the cost of each resource. A fourth is to generate the corresponding cost and revenue values. A fifth is to serve as an intermediate step in building more sophisticated models with economic applications to be used in the simulation models being developed. In particular F-F II can serve as the basis of a product-product model, a model to choose which products and how much of each should be produced to maximize profit.

Assumptions and Capabilities: The model assumes that the production function is a Cobb-Douglass type and that there is only one product. This implies that all of the resources are substitutes for each other and that there is no complementarity between resources. For practical reasons, the number of resources is limited to nine. The number of output levels is limited to 100 by the Dimension Statement. This restriction may be relaxed by changing the size of the dimension and by changing the value of ISAFE in the data statement. ISAFE should be less than or equal to the dimension size. See page 41 for an explanation of ISAFE's purpose.

Algorithm: To generate isoquant fields the user inputs the level of output and the quantities of the various resources. The program changes the quantity of one resource and calculates how much of a second resource is needed holding the quantity of all other resources and the quantity of output constant. This, in effect, is an isoquant. Output is then changed by a specified amount and the calculations repeated, thus generating a second isoquant. This continues until a specified output level is reached. The series of isoquants so generated is called an isoquant field. At this point, the program may terminate or it may introduce changes, specified by the user, in the parameters, and regenerate up to thirty times for a total of thirty-one fields. The first level of output is Y. The change in output is CHY and the highest level of output is YLAST.

If an expansion path is generated,only one point on each isoquant (each Y value) can be the cost minimizing quantity and thus there is only one combination of quantities of resources which are chosen for each Y value. This is still valid even if it is only a partial expansion path. (If some resources are held constant).

The quantities of the resources are X1, X2,...Xn, where n is less than or equal to 9. One resource is specified as the independent resource X(INDE) by the user (see Input File). The program replaces the quantities of all other resources in the production function by quantities of the independent resource such that the Marginal Rate of Technical Substitution (MRTS)

of each of these other resources with the independent resource is equal to their price ratios. In effect, this reduces the production function to a function of one variable, the independent resource. Thus Y becomes a function of X(INDE), the independent resource. Using the specified quantity of X(INDE) the level of output is calculated. This level of output is compared with the specified level of output and if they are not equal then X(INDE) is increased by a proportion such that when Y is again calculated they will be equal. The result at this point is that X(INDE) is now at the cost minimizing quantity for the specified level of output. It is only necessary now to calcualte the quantities of the other resources. This is done by setting the quantity of each of the other resources equal to X(INDE) times given ratios such that the MRTS of each of the other resources with X(INDE) is equal to the ratio of the price of X(INDE) and the price of the resource. This approach is carried out in a sub-routine called "COSMIN" and the user should consult the flow-chart for "COSMIN." The basic principle of this technique is to work in terms of only one resource, X(INDE) instead of 2 to 9 resources and then once the cost minimizing quantiy of X(INDE) is known to calculate the appropriate quantities of the other resources in terms of X(INDE).

The reader will wish to check this procedure. The vital point is whether the MRTS between each resource and every other resource is equal to the price ratios of the two resources. The MRTS and price ratios can be checked by looking at the matrix of these values generated in the output. (see pages 68-69). Since there may be up to nine resources, up to

eighty-one different MRTS values and price ratios are compared. The crucial issue is that the quantity of each resource not specified can be calculated so that the MRTS between each of these resources and X(INDE) is equal to the inverse ratio of the price of the resource to the price of X(INDE) and that these quantities result in a MRTS between any two resources not given or specified such that it is equal to the inverse of the ratios of the prices of these two resources. This is demonstrated with the output matrix (see pages 68-69) providing an emperical verification of the mathematical description discussed later.

The program will also generate other types of tables. One is a listing of the cost and revenue values at each level of output (see pages 70-71). To get cost, the program will first choose the minimum cost quantity of each resource and multiply that quantity by the price of the resource. The product of this result for each variable resource will be summed to get total variable cost and from this average variable cost will be calculated. A similar procedure will be used for any resources held constant and their cost added to get total fixed cost and average fixed cost. It is important to note that while the amount of total cost at each output level is obtained by this means, a true total cost function TC = f(Y), is not derived. Thus it is impossible to take the first derivative of TC and label this marginal cost. To get marginal cost (MC) the program calculates TC at two output levels about each level of Y. These output levels are Y + Range and Y - Range. Range is a small change in output. It may be specified by an assignement statement or if it is not then Range = 1.0.

The program takes the difference between these two output levels and divides this difference by 2 Range. The result is labeled MC.

The program will also generate a matrix of the marginal rates of technical substitution and price ratios between each resource and every other resource. This is vital in demonstrating that the quantities of the resources are actually minimum cost quantities. The number of resources is currently limited to nine as this is the number which will fit on a computer printout page with one column left for the level of output.

Mathematical Description: The Cobb-Douglass production function is of the form:

$$Y = A X_1^{B_1} X_2^{B_2} \ldots X_n^{B_n}$$

where X_i is the quantity of the i'th resource, Y is the quantity of output, A and B_i are the parameter estimates and $n \leq 9$. The marginal rate of technical substitution, MRTS, is the ratio of the marginal productivity of two resources holding all other resources constant. The marginal productivity of a resource, MP, is the partial derivative of the production function with respect to that resource. The marginal rate of substitution, MRS, of one resource for the other is the ratio of the marginal product of each resource.

Thus in the general form:

$$MRTS_{X_j X_i} = \frac{MP_j}{MP_i} = \frac{B_j X_i}{B_i X_j}$$

Similarly at cost minimization:

$$MRTS_{X_{(inde)}X_j} = \frac{MP_{(inde)}}{MP_j} = \frac{B_{(inde)}X_j}{B_jX_{(inde)}} = \frac{P_{(inde)}}{P_j}$$

[where (inde)refers to the independent (or specified) resource and P is price.].

Rearranging terms it can be shown that:

$$X_j = \frac{X_{(inde)}B_jP_{(inde)}}{B_{(inde)}P_j}$$

also:

$$X_i = \frac{X_{(inde)}B_iP_{(inde)}}{B_{(inde)}P_i}$$

Next calculate the constant, K

$$K = \frac{X_{(inde)}P_{(inde)}}{B_{(inde)}}$$

then solve for X_i and X_j in terms of this constant,

$$X_i = \frac{B_i}{P_i} K$$

$$X_j = \frac{B_j}{P_i} K$$

and rearranging terms we have:

$$K = \frac{X_iP_i}{B_i} = \frac{X_jP_j}{B_j}$$

Thus the general equilibrium condition is met:

$$\frac{P_i}{P_j} = \frac{X_j B_i}{X_i B_j} = MRTS_{X_i X_j} = \frac{MP_i}{MP_j}$$

For partial equilibrium, any combination of resource levels may be specified or held constant except one, i.e., one or more must be variable.

References:

[1] James Wherly , Marginal Analysis, undated documentation

[2] Stanley Wilson and Ray Billingsley, FACTOR-FACTOR I Program Documentation, October 1970.

2. PROGRAM DESCRIPTION

Program Abstract: The main program inputs and prints the data, prints the isoquants and introduces changes in the values of the parameters of the model. The variables inputed are described in the Input File.

Variable List: The variables of the production function have been identified in the Data requirements. Most variables have roles which are relevant to a particular subroutine and are explained in that subroutine's description. Variables such as Y1, X1(I) and CHY1 are used to reinitialize Y, X(I) and CHY to their original values in preparation for changing the functions parameters. Variables such as BSUM, SUMNOO, NOODLE(I), R, YSUM(I), and CON are used as intermediate variable names to simplify a sequence of calculations.

Subroutines/Functions Called: The main program does not call any functions but it calls several subroutines. These are:

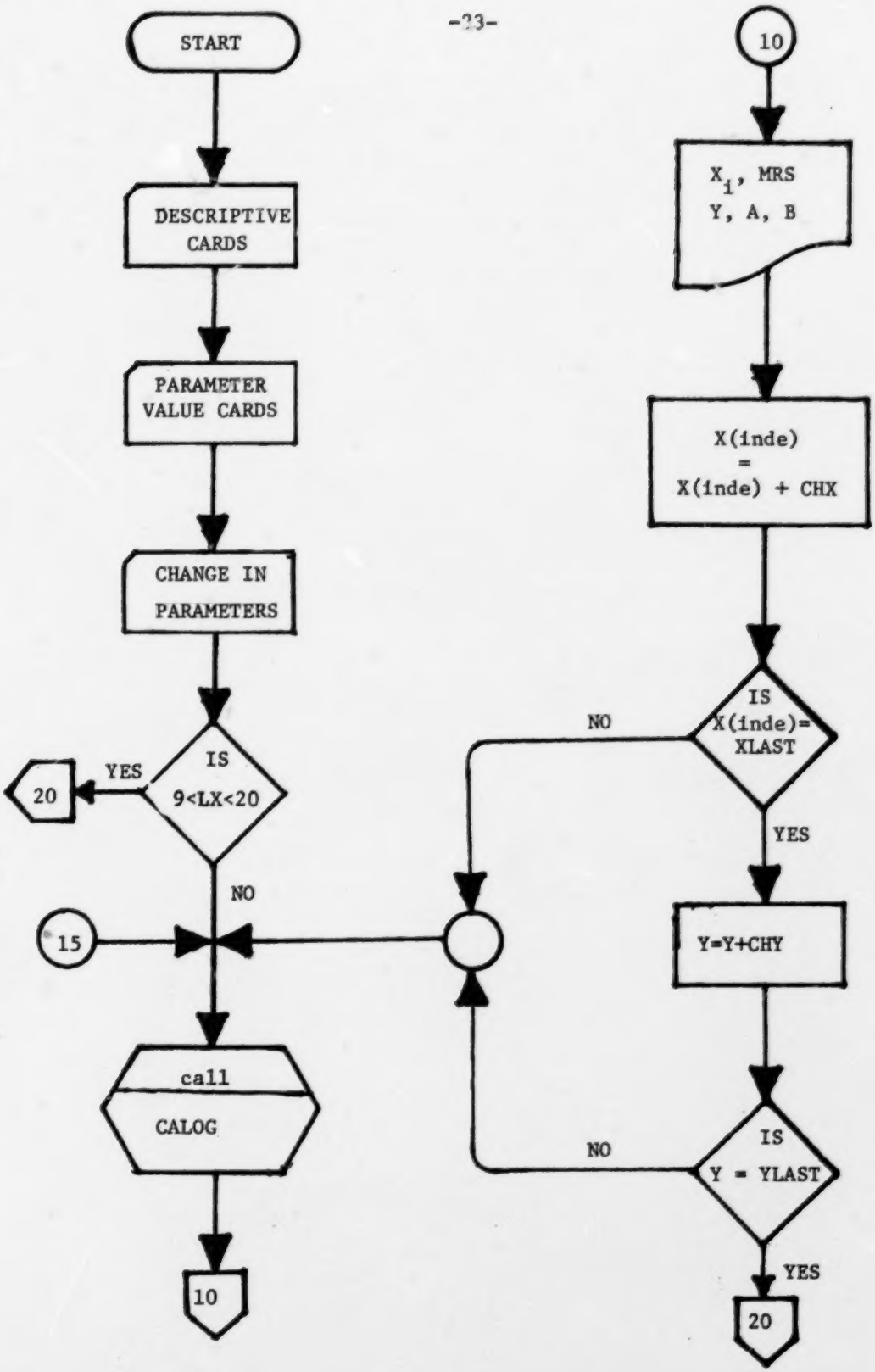
1. ASIDE
2. VMPOUT
3. COSOUT
4. CALOG
5. PTRIX
6. CALY
7. CALVMP
8. COSMIN
9. ZEROIN

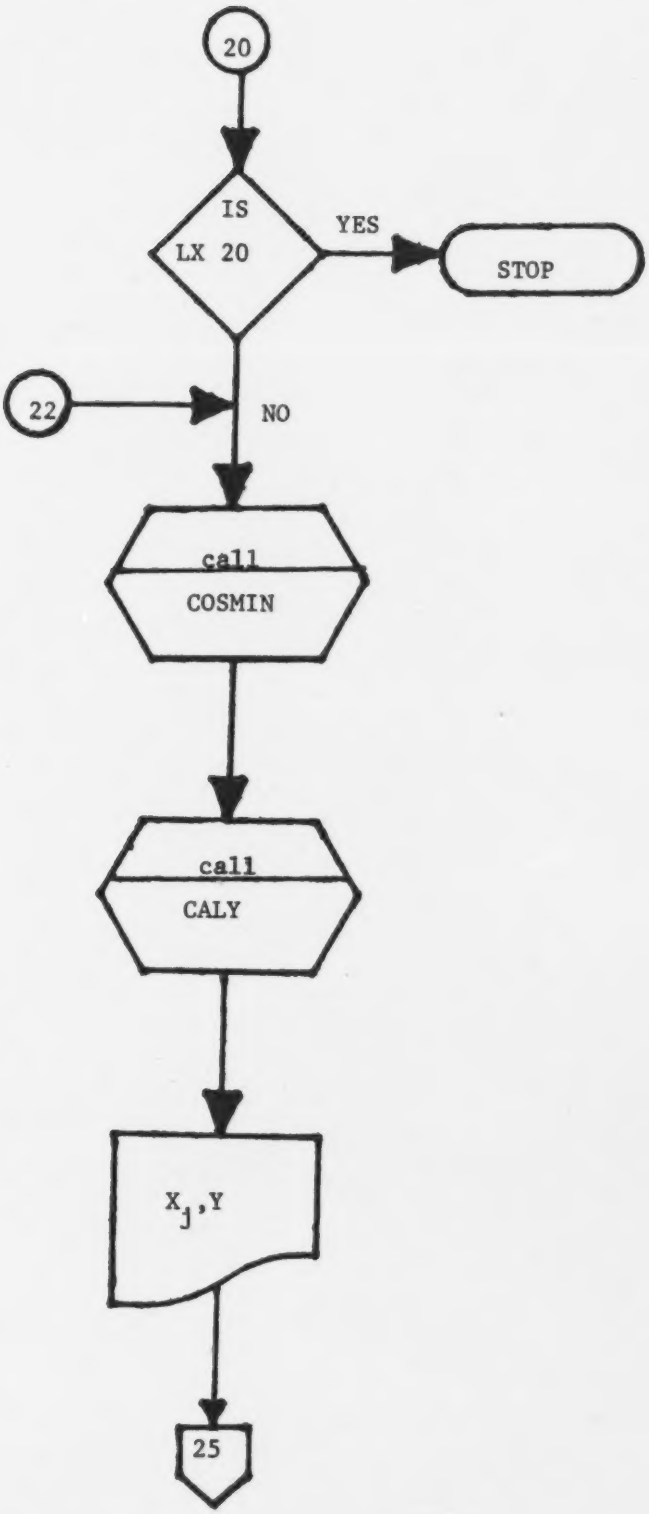
The user should refer to the program description of each of these. Three of these subroutines perform straightforward functions and will be described here instead of having separate sections devoted to them.

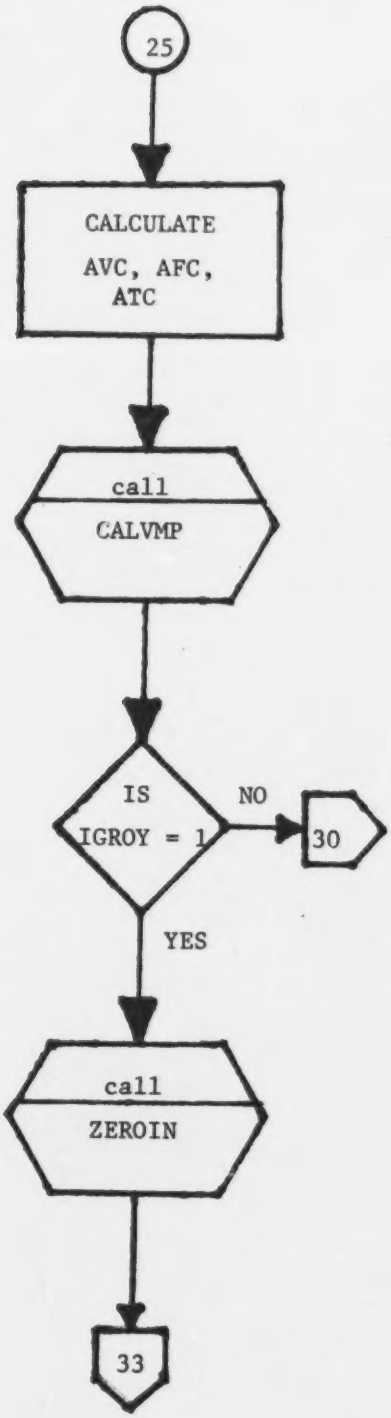
Subroutine ASIDE calls in values for IA and A and returns values for L10A, LEA and AREAL. If IA is 10, the program interprets A as the actual value of A, (AREAL) and proceeds to calculate L10A, (log base 10 of AREAL) and LEA (log base e of AREAL). If IA is greater than 10, the program sets L10A equal to A and uses this to calcualte LEA and AREAL. If IA is less than 10 the program sets LEA equal to A and uses this to calculate L10A and AREAL.

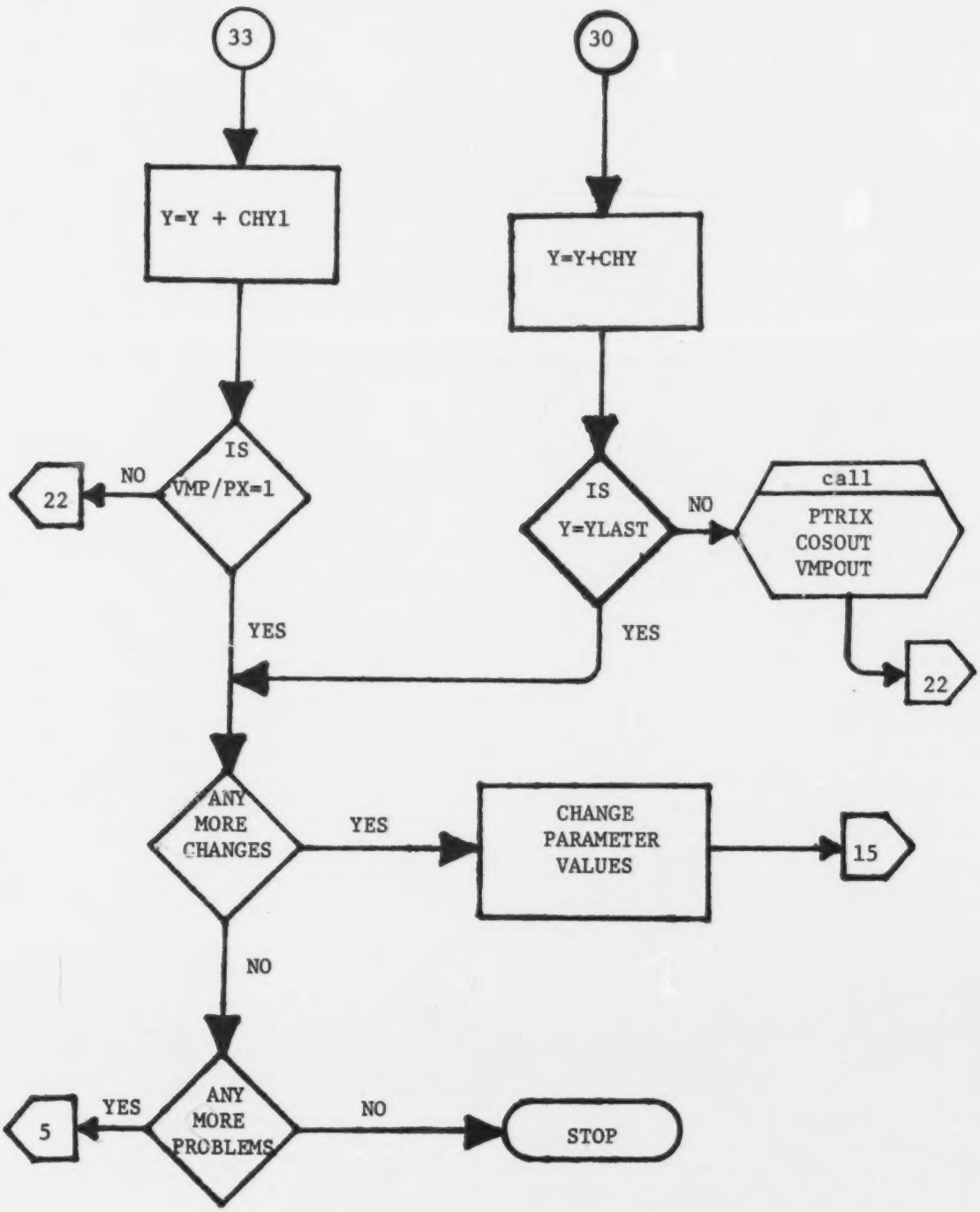
Subroutine COSOUT prints out the values for the COST FUNCTION table. These are level of output (Y), total cost (TC), total variable cost (TVC), average total cost (ATC), average fixed cost (AFC), average variable cost (AVC), marginal cost (MC), marginal revenue (MR), total revenue (TR), and profit. The subscripted variables are TFC, TVC, TC, AVC, AFC, ATC, TR, AR, MC and SUBY. SUBY is the value of Y at each itteration. The subscript is ICC and the write statement is in a DO Loop with ICSTOP used to terminate it. The parameters of the model, B, PX, L1OA, LEA, AREAL, and FC are called in to be printed out at the top of the table. DES is called in. It is subscripted and is the name of the character information on the last description card. COSOUT returns no values.

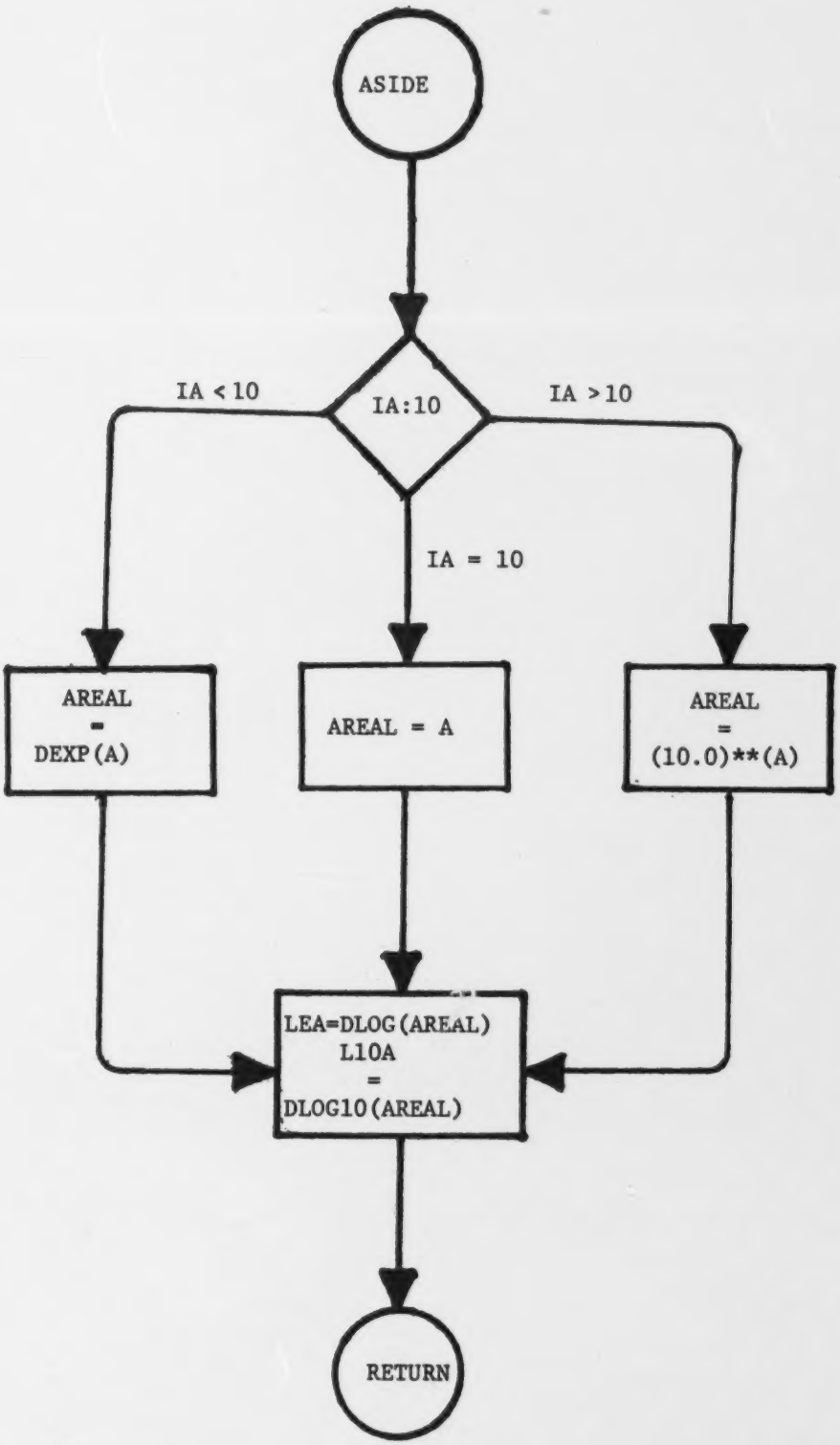
Subroutine VMPOUT has a function analogical to that of COSOUT. It calls in the marginal product of X(I), MPX(I), the value of the marginal product of X(I), VMPX(I), and the ratio VMP(I)/PX(I), which is PMPR(I). In addition it calls in Y, PX, B, AREAL, L1OA, LEA, ICC, DES, SUBY, ICSTOP and MR. It returns no values. VMPOUT prints the values of PMPR and MPX. B, PX, L1OA, LEA and AREAL are printed on the top of the table. This table is labeled VMPX/PX and MP. When PMPR = 1.0, output is at the profit maximizing quantity. VMPOUT returns no values. Neither ASIDE, COSOUT, or VMPOUT call any function or subroutine.

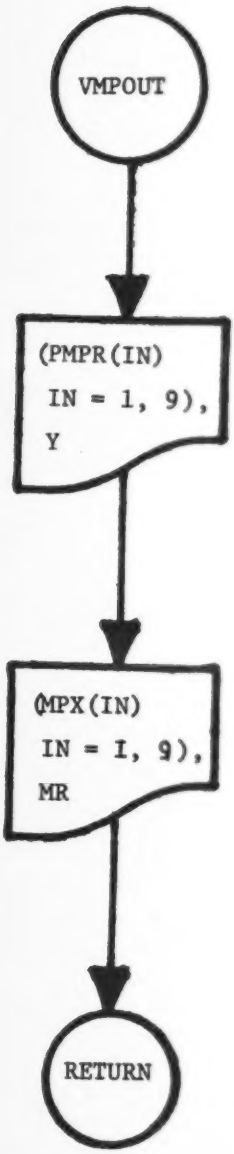

START
DESCRIPTIVE CARDS
PARAMETER VALUE CARDS
CHANGE IN PARAMETERS
IS 9<LX<20
YES
20
NO
15
call CALOG
10
10
X_i, MRS Y, A, B
X(inde) = X(inde) + CHX
IS X(inde)= XLAST
NO
YES
Y=Y+CHY
IS Y = YLAST
NO
YES
20

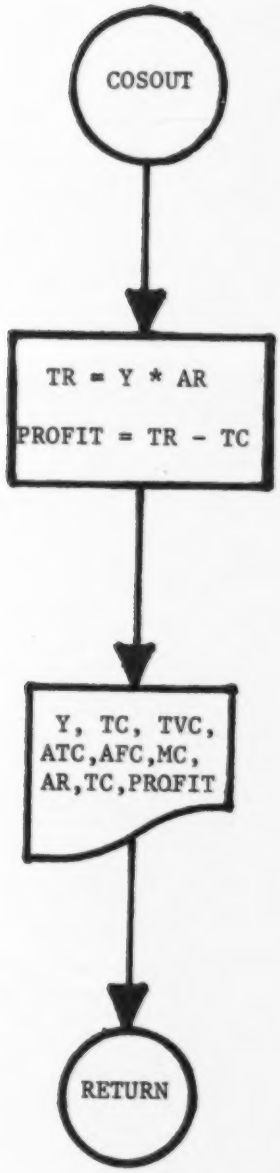

20
IS
LX 20
YES
STOP
22
NO
call
COSMIN
call
CALY
X_j,Y
25


25
CALCULATE
AVC, AFC,
ATC
call
CALVMP
IS
IGROY = 1
NO
30
YES
call
ZEROIN
33


33
30
Y=Y + CHY1
Y=Y+CHY
IS
VMP/PX=1
NO
22
IS
Y=YLAST
NO
call
PTRIX
COSOUT
VMPOUT
YES
YES
22
ANY
MORE
CHANGES
YES
CHANGE
PARAMETER
VALUES
15
NO
ANY
MORE
PROBLEMS
YES
5
NO
STOP


ASIDE
IA:10
IA < 10
IA = 10
IA > 10
AREAL = DEXP(A)
AREAL = A
AREAL = (10.0)**(A)
LEA=DLOG(AREAL)
L10A = DLOG10(AREAL)
RETURN

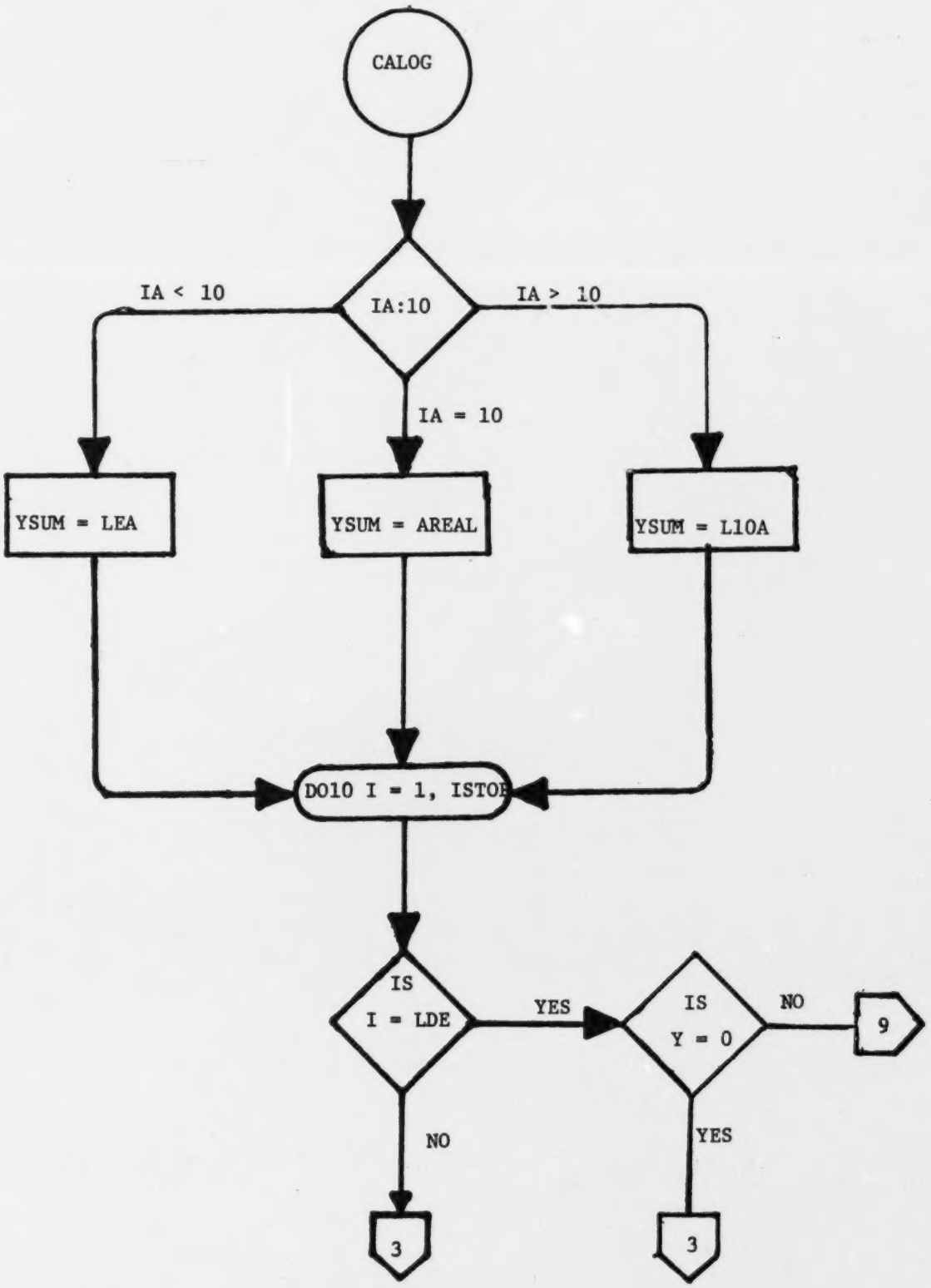

VMPOUT
(PMPR(IN)
IN = 1, 9),
Y
(MPX(IN)
IN = I, 9),
MR
RETURN

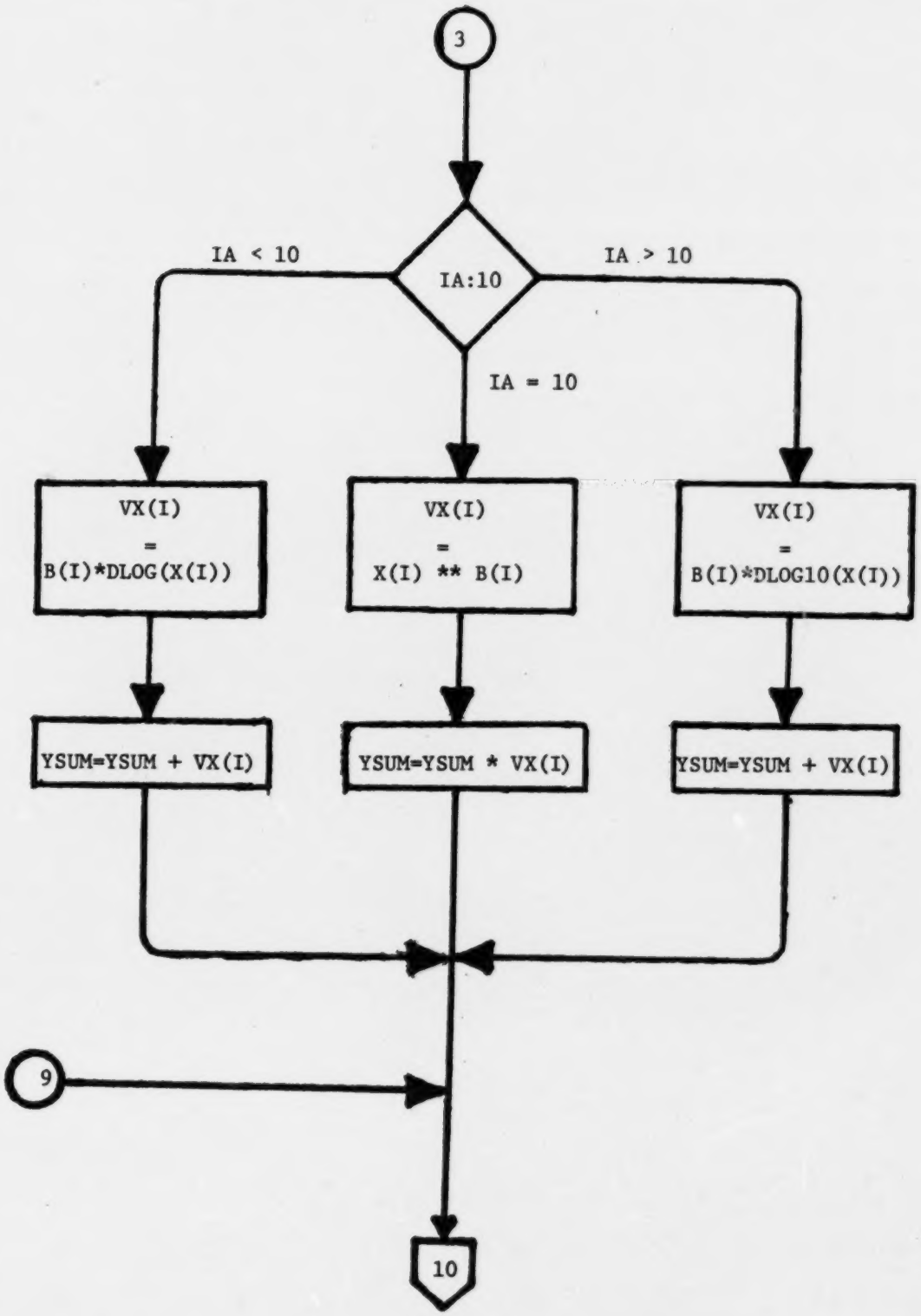

COSOUT
TR = Y * AR
PROFIT = TR - TC
Y, TC, TVC,
ATC,AFC,MC,
AR,TC,PROFIT
RETURN

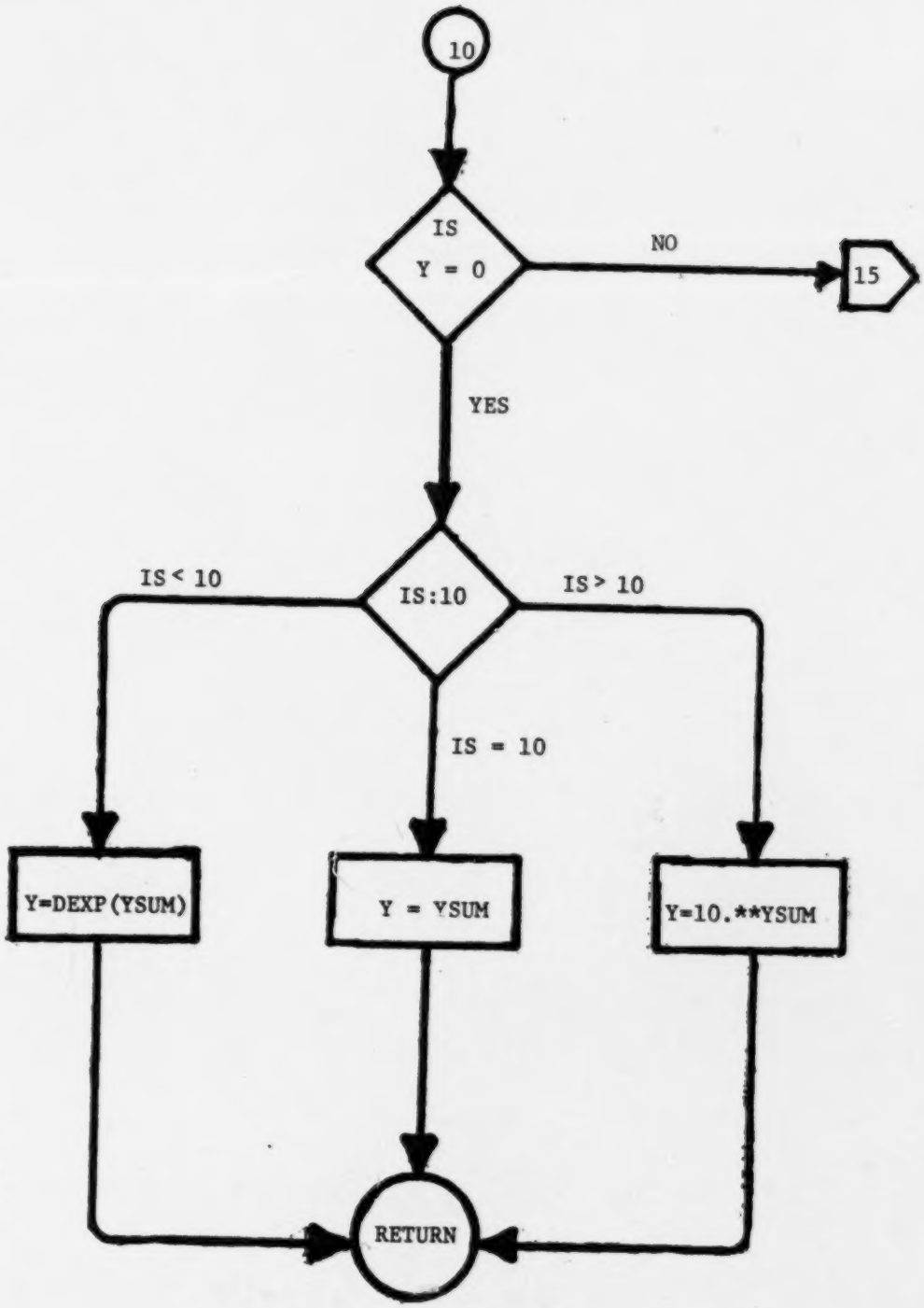
SUBROUTINE CALOG

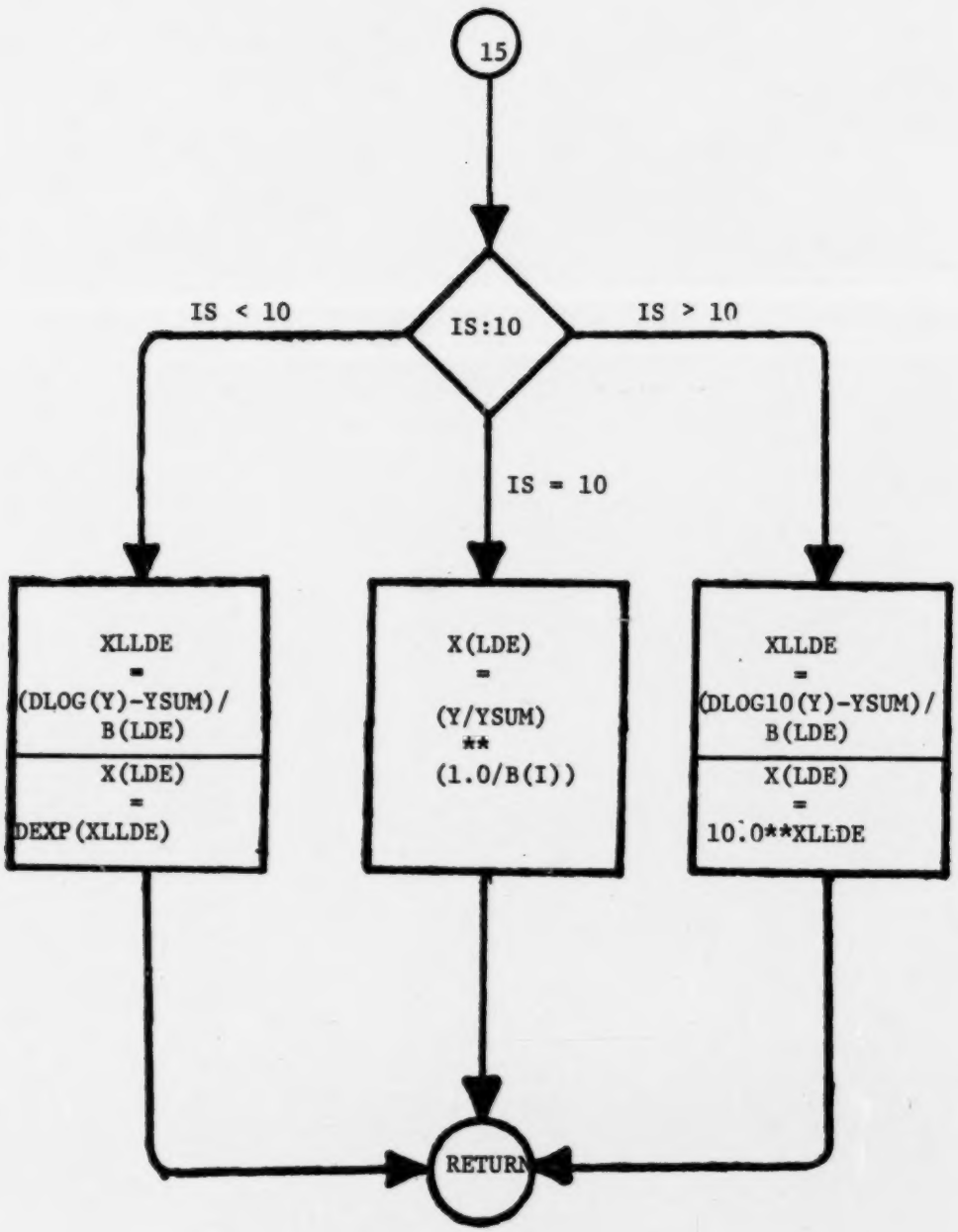
ABSTRACT: This subroutine is used to calculate isoquants. In effect, it calculates X(LDE) the dependent resource. It calls values of A, Y, ISTOP, IX, B, J, KELX, N, Y1, L10A, LEA, IA, LDE and AREAL and returns a value for X(LDE). Depending upon the value of IA, CALOG will calculate X(LDE) in three different ways, summing the logs base 10 of X(I), the logs base e of X(I) or taking the product of all X(I)'s except X(LDE). The subroutine also decides whether the first value of Y or X(LDE) should be calculated depending on whether or not a value for Y is entered on the Y card or left blank. CALOG is one of the subroutines which must be changed if another production function besides the Cobb-Douglas is used. CALOG calls no other subroutines or functions.

An additional complication of CALOG is that this subroutine must calculate an original value of output if Y is left blank. If Y is zero, then the log of each X(I), including X(LDE), must be summed up and then the anti log of this sum taken. The subroutine follows this procedure if it finds that Y is zero. If IA is 10 then it takes the product of all X(I) ** B(I) including X(LDE) ** B(LDE). After the original value of Y is calculated, however, it is then necessary to calculate all subsequent values of X(LDE). This implies that VX(I) (where VX(I) = X(I) ** B(I) or VX(I) = B(I) * logX(I)) will not be calculated except under the double condition that I is not equal to LDE and that Y, is not equal to zero.


CALOG
IA:10
IA < 10
IA = 10
IA > 10
YSUM = LEA
YSUM = AREAL
YSUM = L10A
DO10 I = 1, ISTOP
IS
I = LDE
YES
IS
Y = 0
NO
9
NO
YES
3
3


3
IA:10
IA < 10
IA = 10
IA > 10
VX(I) = B(I)*DLOG(X(I))
VX(I) = X(I) ** B(I)
VX(I) = B(I)*DLOG10(X(I))
YSUM=YSUM + VX(I)
YSUM=YSUM * VX(I)
YSUM=YSUM + VX(I)
9
10

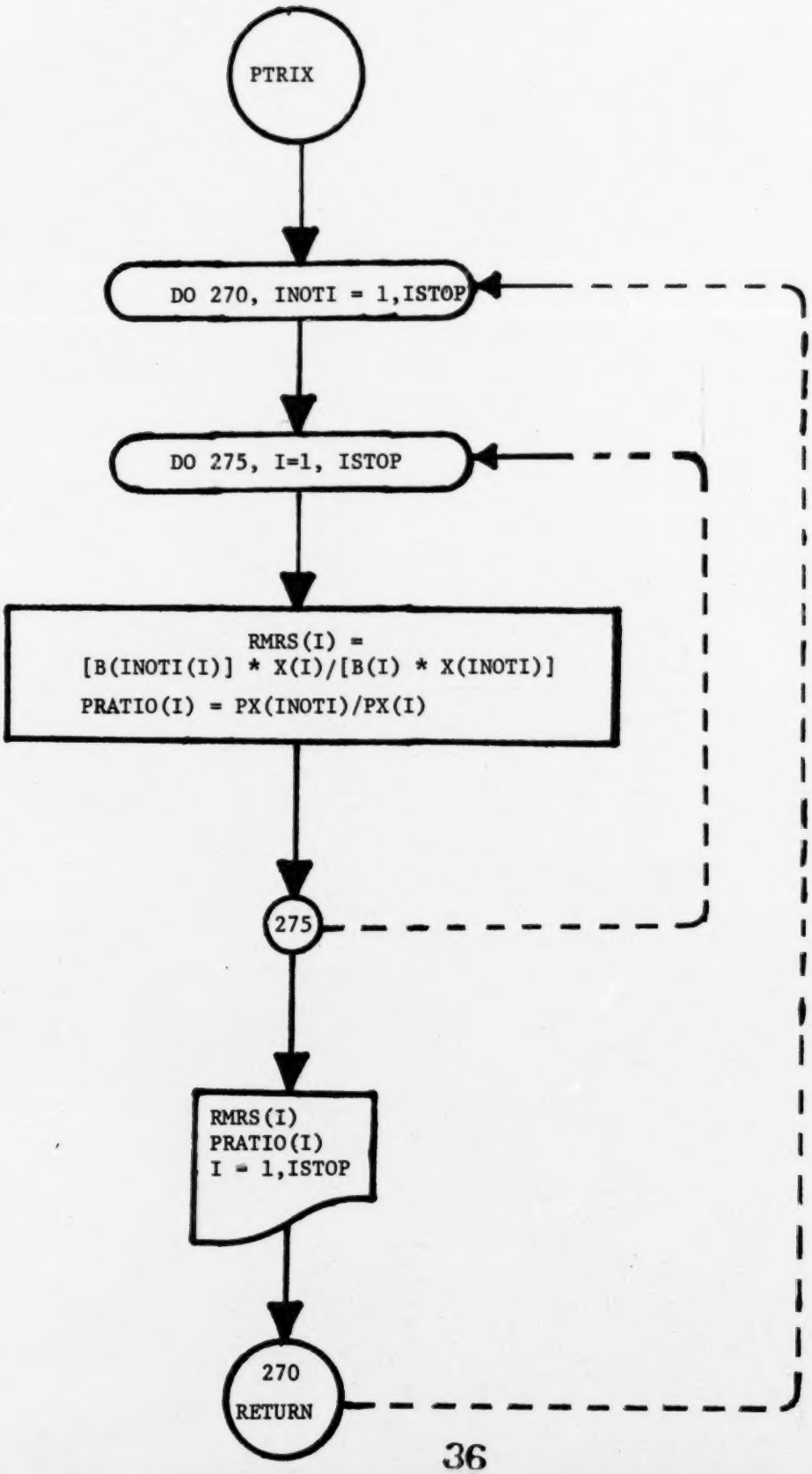

10
IS
Y = 0
NO
15
YES
IS:10
IS < 10
IS > 10
IS = 10
Y=DEXP(YSUM)
Y = YSUM
Y=10.**YSUM
RETURN


15
IS:10
IS < 10
IS = 10
IS > 10
XLLDE = (DLOG(Y)-YSUM)/ B(LDE)
X(LDE) = DEXP(XLLDE)
X(LDE) = (Y/YSUM) ** (1.0/B(I))
XLLDE = (DLOG10(Y)-YSUM)/ B(LDE)
X(LDE) = 10.0**XLLDE
RETURN

SUBROUTINE PTRIX

ABSTRACT: This subroutine is used to calculate and print out the Matrix of the MRTS and Price Ratio values for each pair of resources. It calls in DES, Y, B, AREAL, L10A, LEA, PX, ISTOP and X. The MRTS in the subroutine has the variable name RMRS(I) and the price ratio has the variable name PRATIO(I). PTRIX returns no variable values. The fact that RMRS and PRATIO are names not used in other parts of the program and are not arguments in the subroutine call statement assures the user that they are determined solely by the values of X(I) called in and thus are an independent test that the cost minimization quantities of X(I) have been calculated. The table is labeled in the output, MRTS-PRICE RATIO MATRIX. PTRIX is one of the subroutines which must be changed if another production function is substituted for the Cobb-Douglas in the program. It calls no other functions or subroutines.

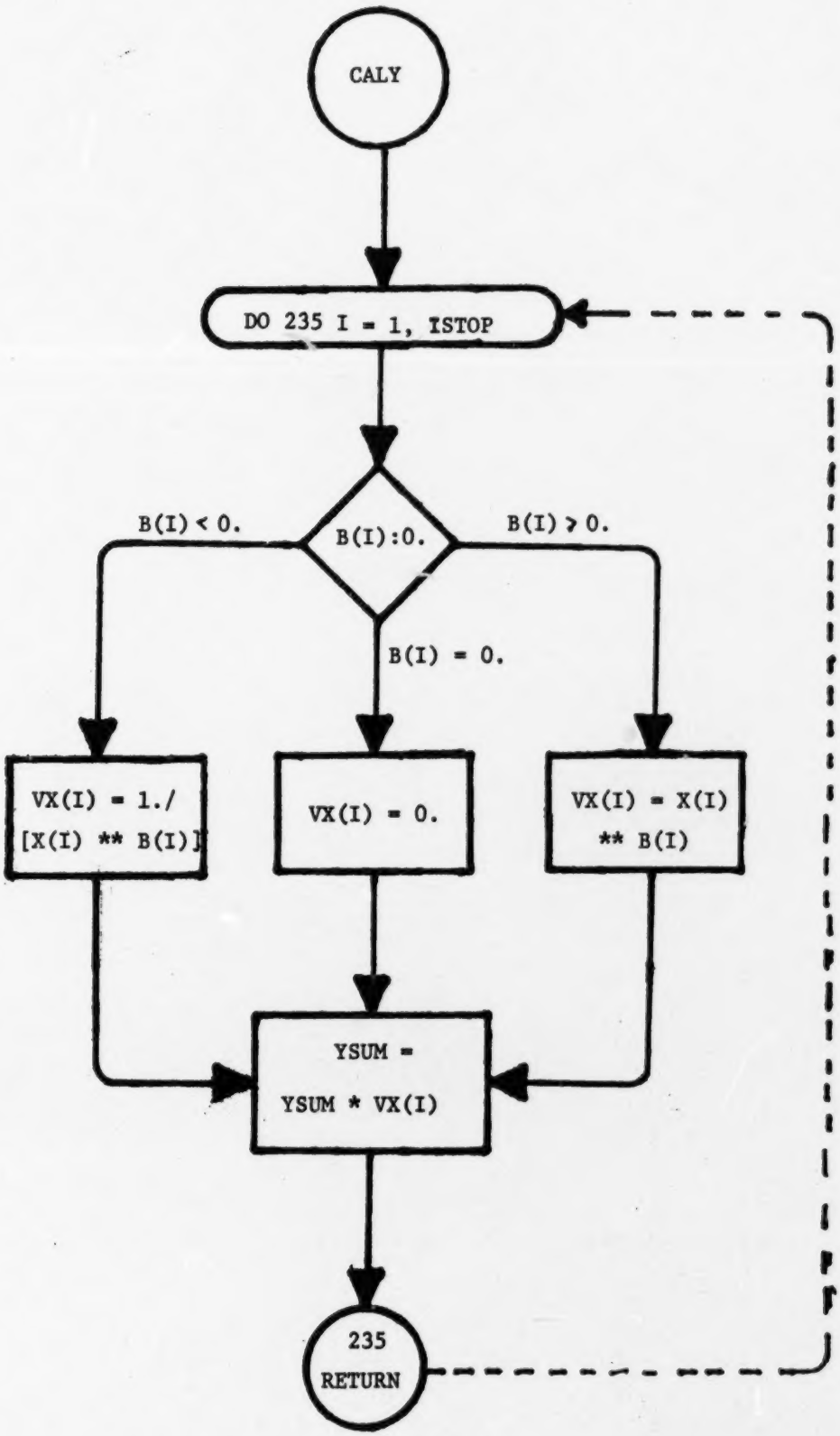
The basic technique of PTRIX is the nested DO loops. The inner DO loop generates the MRTS and price ratio between a given resource and each resource. These resources are taken in turn because the inner DO loop increments the subscript I. But the subscript of the given resource must not be changed and thus its subscript name must be different from I. The name INOTI accomplishes this. But each resource must become the given resource and this is accomplished by the outer DO loop incrementing INOTI.


PTRIX
DO 270, INOTI = 1,ISTOP
DO 275, I=1, ISTOP
RMRS(I) =
[B(INOTI(I)] * X(I)/[B(I) * X(INOTI)]
PRATIO(I) = PX(INOTI)/PX(I)
275
RMRS(I)
PRATIO(I)
I = 1,ISTOP
270
RETURN

SUBROUTIONE CALY

ABSTRACT: This subroutine is used to calculate a value of Y. It calls in B, X, ISTOP and AREAL and returns a value of Y. Y is set equal to the product of AREAL and VX(I), where I = 1,...ISTOP. In general, VX(I)=X(I)**B(I), but the subroutine must first examine the value of B(I) to determine whether it is zero, negative or positive. This will determine exactly how VX(I) is calculated.

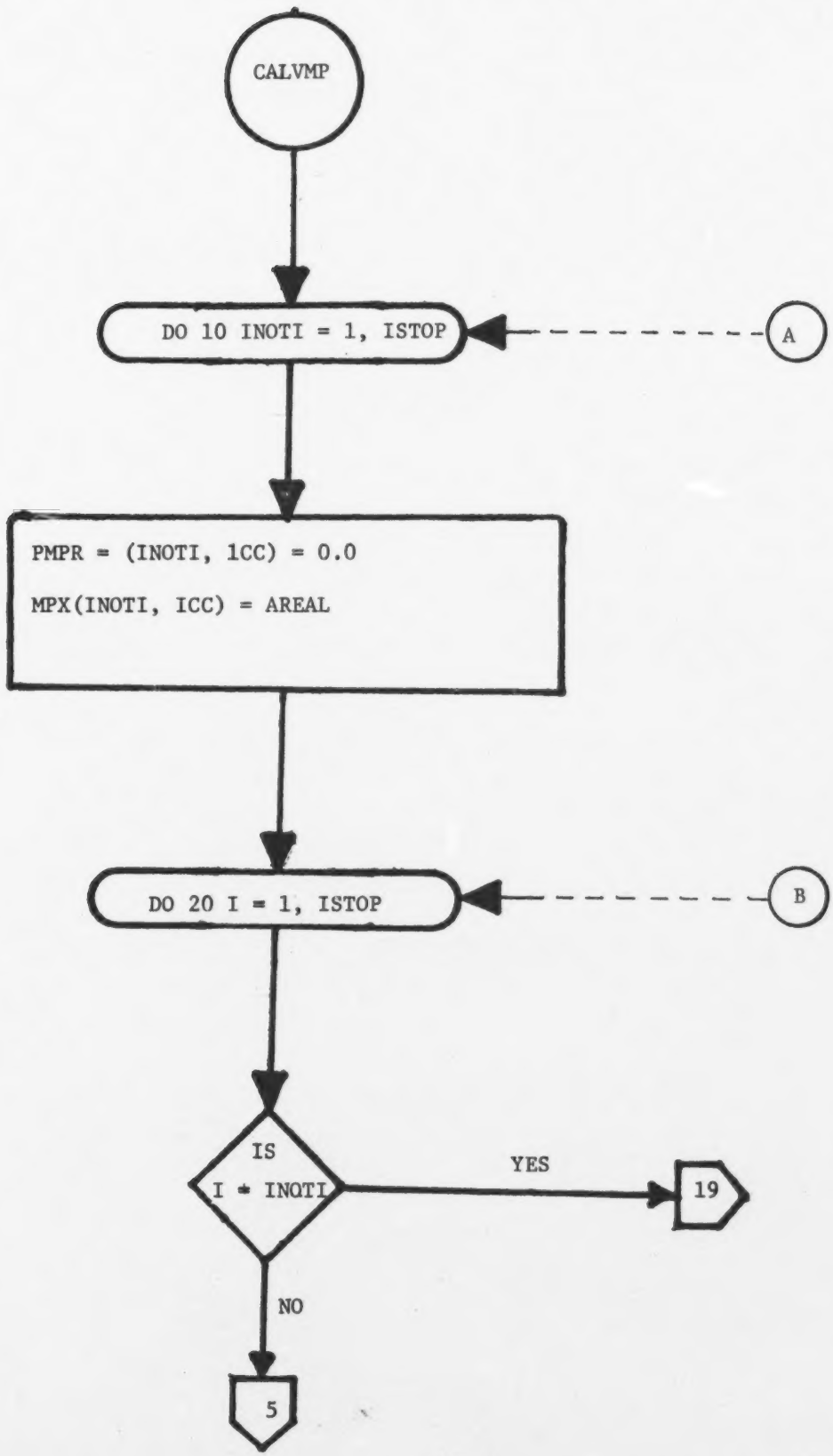
CALY is called just after COSMIN. COSMIN selects the cost minimizing quantitites of each resource. This determines the values of X(I). Then CALY uses these values to again calculate Y so that the user is assured these values of X(I) will indeed produce the specified value of Y. Then X(I) and Y are printed in the next statement so that this is verified in the printout. This table is labeled EXPANSION PATH-PHYSICAL INPUT & OUTPUT. CALY is one of the subroutines which must be changed if another type of production function is substituted for the Cobb-Douglass in the program. CALY calls no other functions or subroutines.

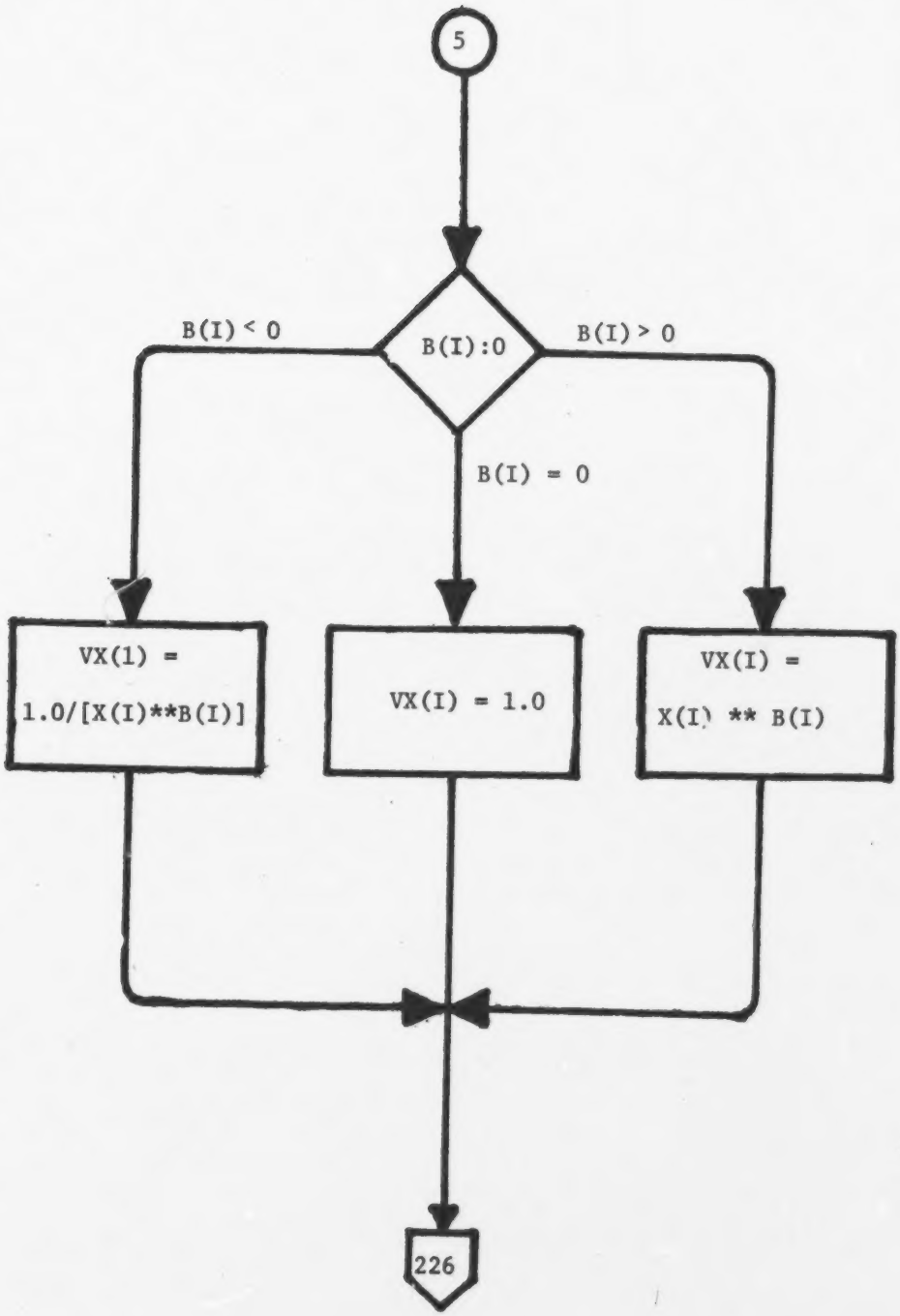

CALY
DO 235 I = 1, ISTOP
B(I) < 0.
B(I):0.
B(I) > 0.
B(I) = 0.
VX(I) = 1./
[X(I) ** B(I)]
VX(I) = 0.
VX(I) = X(I)
** B(I)
YSUM =
YSUM * VX(I)
235
RETURN

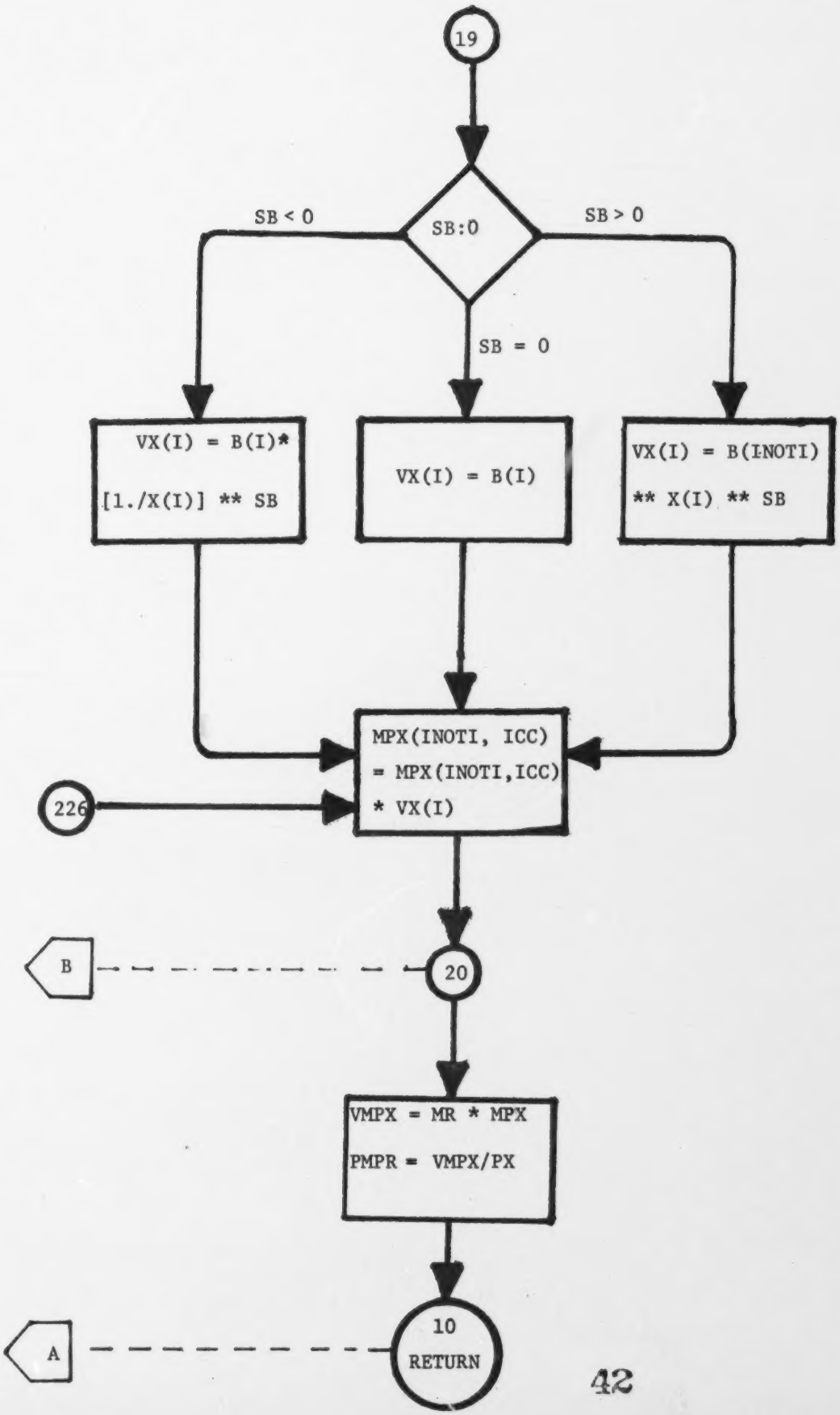
SUBROUTINE CALVMP

ABSTRACT: This subroutine calls in AREAL, X, B, PX, ISTOP, Y, MR, and ICC and uses these to calculate VMPX, PMPR, and MPX. ISTOP is used to terminate DO Loops and is the total number of resources. ICC subscripts VMPX, PMPR and MPS in order to associate them with unique output levels.

Two variables are generated in this subroutine but not returned. They are SB(where SB=B(I)-1.0) and VX(I)(where VX(1)=X(I)**B(I)). The MPX of a given resource is the partial derivative of the production function with respect to that resource. This value is the product of AREAL, and the values of VX(I). In general, for the VX(I) of the given resource, VX(I)=X(I)**SB. But, the subroutine must also examine the values of B(I) and SB to decide whether they are zero, negative or positive. Depending on this decision VX(I) must be calculated in different ways. (See flowchart) This process is repeated for each resource. CALVMP returns values for VMPX, PMPR and MPX. If a different type of production function besides the Cobb-Douglass is used then CALVMP is one of the subroutines which must be changed. CALVMP calls no other functions or subroutines.


CALVMP
DO 10 INOTI = 1, ISTOP
A
PMPR = (INOTI, 1CC) = 0.0
MPX(INOTI, ICC) = AREAL
DO 20 I = 1, ISTOP
B
IS
I = INOTI
YES
19
NO
5

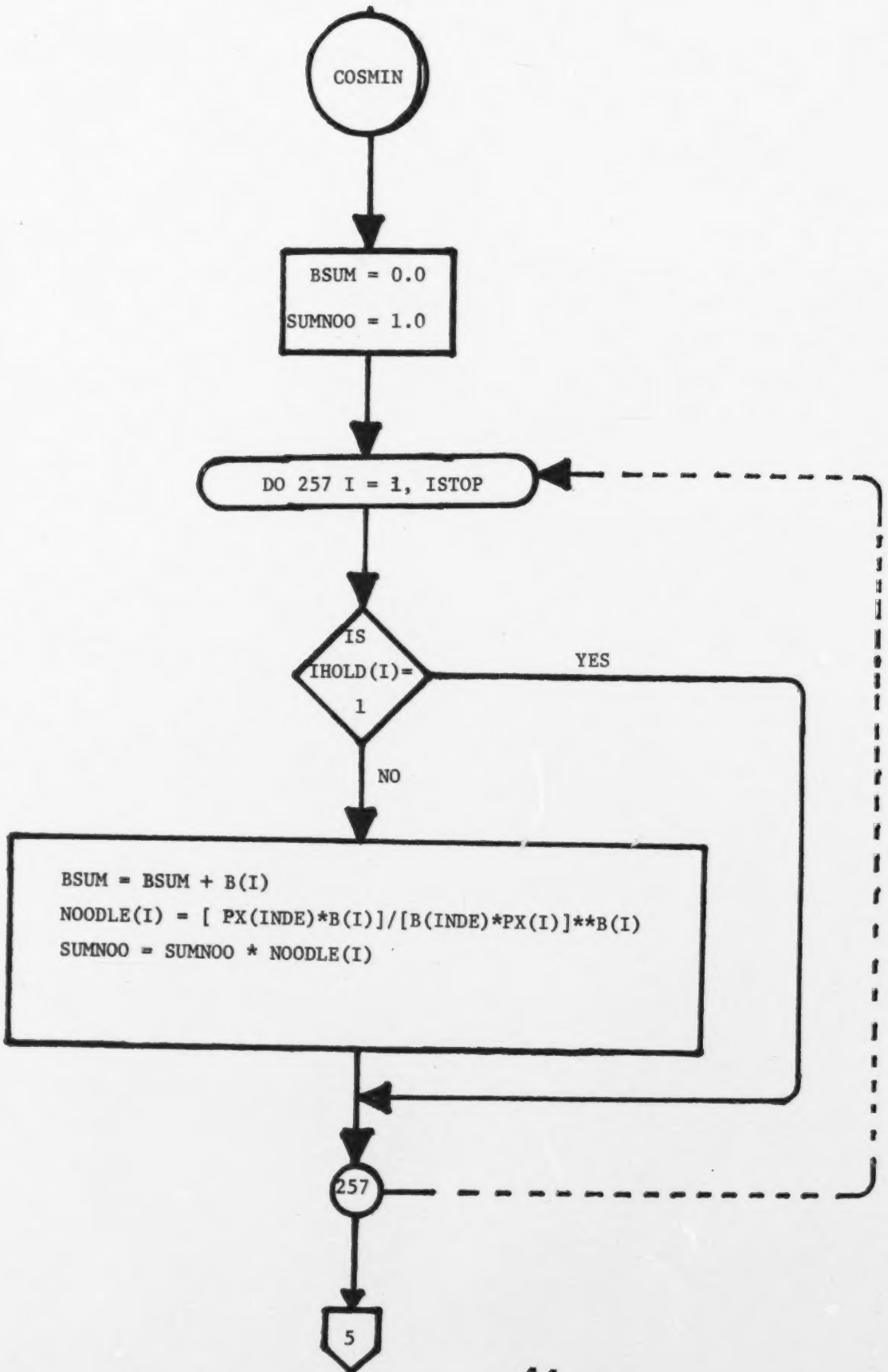

5
B(I):0
B(I) < 0
B(I) = 0
B(I) > 0
VX(1) =
1.0/[X(I)**B(I)]
VX(I) = 1.0
VX(I) =
X(I) ** B(I)
226

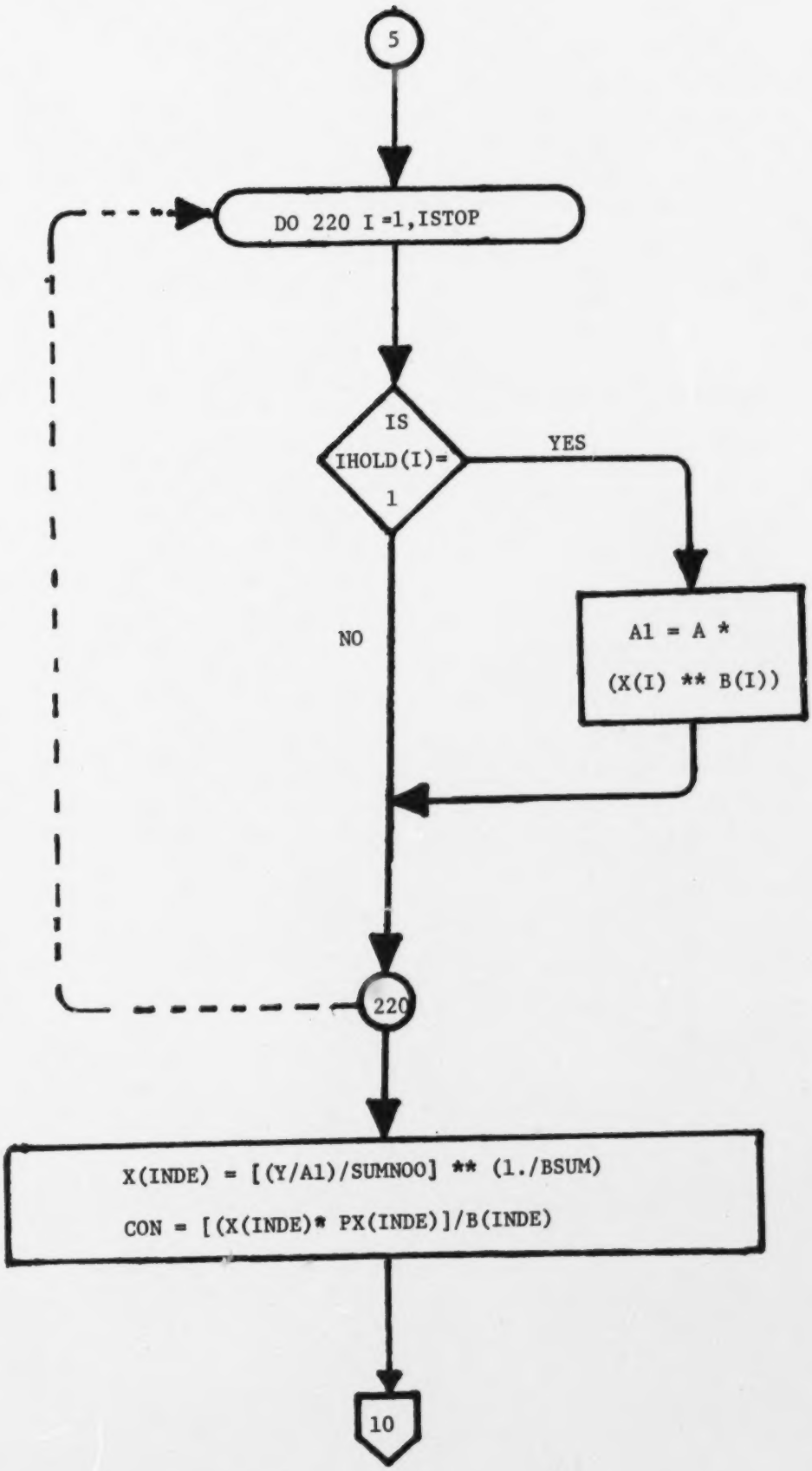

19
SB:0
SB < 0
SB = 0
SB > 0
VX(I) = B(I)*
[1./X(I)] ** SB
VX(I) = B(I)
VX(I) = B(INOTI)
** X(I) ** SB
MPX(INOTI, ICC)
= MPX(INOTI,ICC)
* VX(I)
226
B
20
VMPX = MR * MPX
PMPR = VMPX/PX
A
10
RETURN

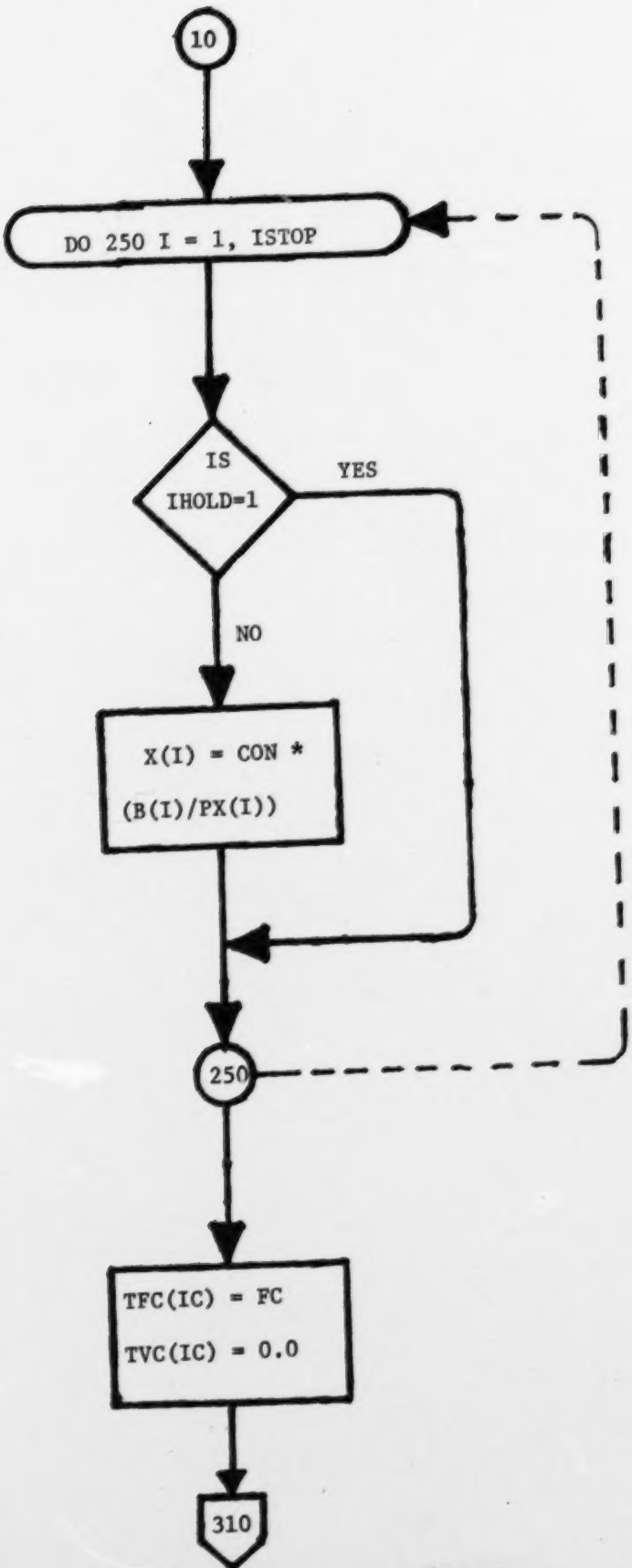
SUBROUTINE COSMIN

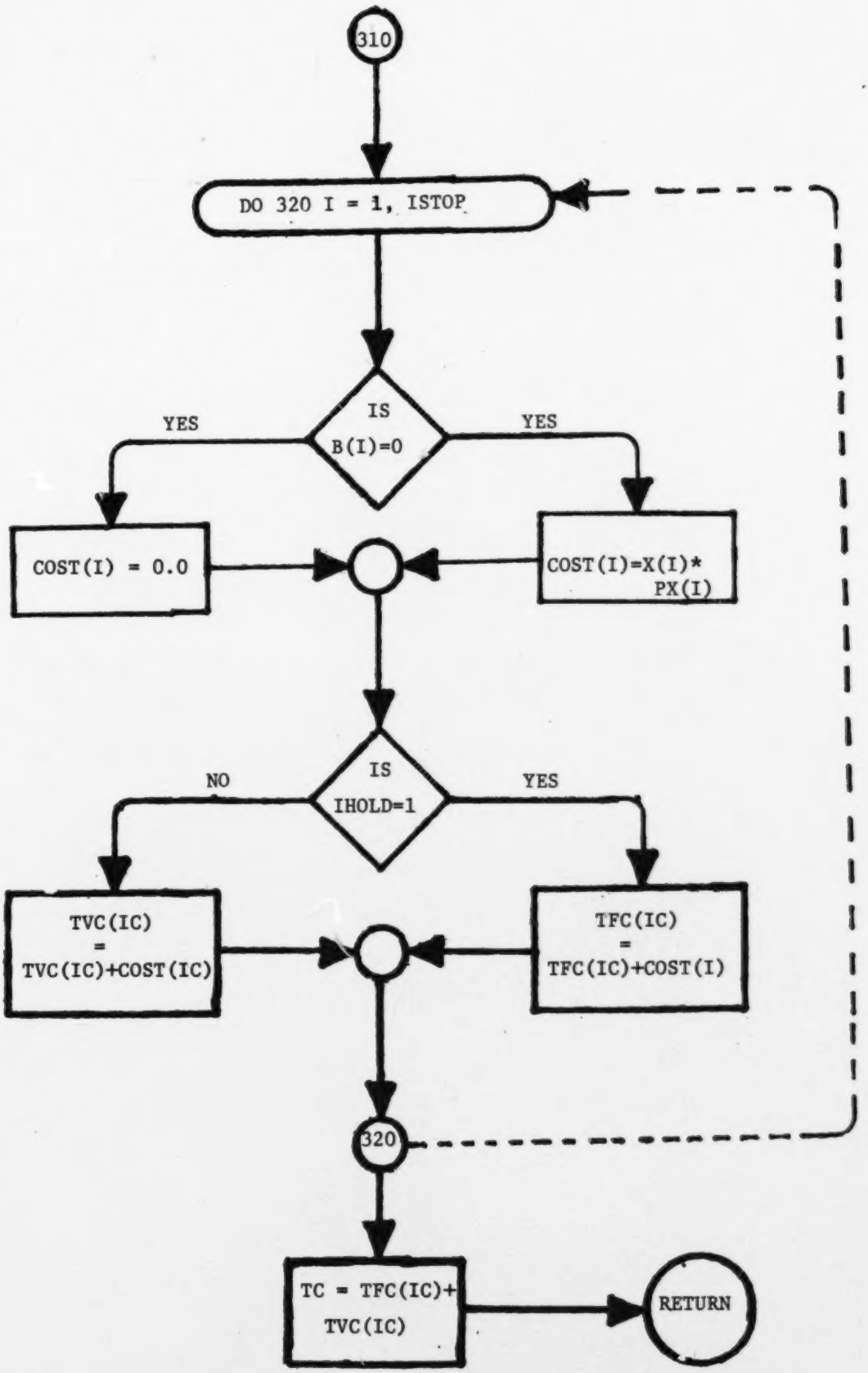
ABSTRACT: This subroutine is used to calculate the minimum cost quantity of each resource, total fixed cost (TFC) and total variable cost (TVC). It calls in Y, B, PX, AREAL, INDE, ISTOP, IHOLD, IC and FC and returns values for X, TFC, TVC, and SUBTC. SUBTC = TFC + TVC. The basic procedure of COSMIN was discussed in the General Description (1.5) and the Mathematical Description. IHOLD(I) is associated with X(I) and was used to determine if the resource is to be held constant or set to its cost minimization level. IHOLD also decides whether Cost(I)(where Cost(I) = X(I) * P(I)) is added to TFC or TVC. COSMIN is one of the subroutines which must be changed if another production function is substituted for the Cobb-Douglass. It calls no other subroutines or functions.

If the user elects to have the program converge on the profit maximization quantity of output the number of changes in Y will not be known for certain. Since many variables are subscripted and these subscripts are increased by one each time Y is increased, it is not unlikely that the program will reach the dimensioned limits of a subscripted variable. When this happens, the run will be terminated before COSOUT, PTRIX and VMPOUT are called. To prevent this, there is a feature in the program which terminates any further increases in Y if the subscript ICC reaches a value equal to the value of ISAFE. (See page 14.) At this point the program calls PTRIX and COSOUT and VMPOUT so that these tables will be printed out and not lost due to the program approaching the dimension limit.


COSMIN
BSUM = 0.0
SUMNOO = 1.0
DO 257 I = 1, ISTOP
IS IHOLD(I)= 1
YES
NO
BSUM = BSUM + B(I)
NOODLE(I) = [PX(INDE)*B(I)]/[B(INDE)*PX(I)]**B(I)
SUMNOO = SUMNOO * NOODLE(I)
257
5

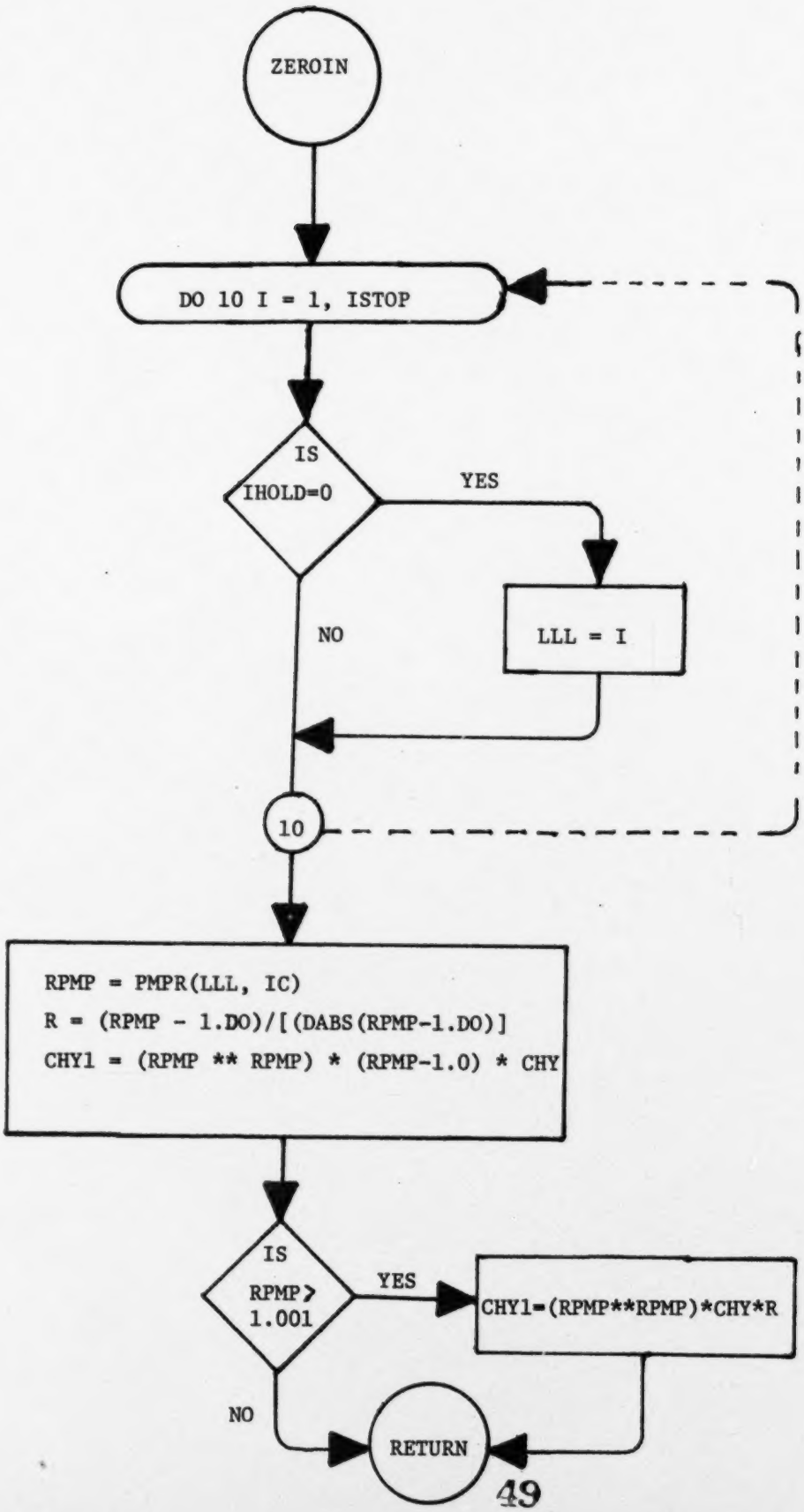

5
DO 220 I =1,ISTOP
IS
IHOLD(I)=
1
YES
NO
A1 = A *
(X(I) ** B(I))
220
X(INDE) = [(Y/A1)/SUMNOO] ** (1./BSUM)
CON = [(X(INDE)* PX(INDE)]/B(INDE)
10


10
DO 250 I = 1, ISTOP
IS
IHOLD=1
YES
NO
X(I) = CON *
(B(I)/PX(I))
250
TFC(IC) = FC
TVC(IC) = 0.0
310


310
DO 320 I = 1, ISTOP
IS
B(I)=0
YES
YES
COST(I) = 0.0
COST(I)=X(I)*
PX(I)
IS
IHOLD=1
NO
YES
TVC(IC)
=
TVC(IC)+COST(IC)
TFC(IC)
=
TFC(IC)+COST(I)
320
TC = TFC(IC)+
TVC(IC)
RETURN

SUBROUTINE ZEROIN

ABSTRACT: This subroutine is used to calculate values for CHY1 such that Y will converge on the profit maximizing value of output. It is called only if the user selected the option which causes Y to converge on the profit maximizing level of output. To converge on this level, the program compares PMPR with 1.0. As Y gets closer to the profit maximization level, PMPR gets closer to 1.0. ZEROIN calculates CHY1 as a function of (PMPR - 1.0). Thus as PMPR approaches 1.0, CHY1 approaches 0.0. ZEROIN also selects the subscript of PMPR such that it is the PMPR of a resource which is not held constant. ZEROIN calls values for Y, PMPR, CHY, ISTOP and IC and returns a values for CHY1. If a user wishes to call ZEROIN he might first consider running his data without calling it, in order to scan a range of output levels and locate the profit maximizing level of output within a narrow range. Then he can run the data, this time calling ZEROIN, and setting Y equal to the output level of one boundary of this range. In this way he can avoid having the program produce a large number of output levels but never reach the profit maximizing level before the subscript of PMPR reaches its dimension limit. ZEROIN calls no other function or subroutine.


ZEROIN
DO 10 I = 1, ISTOP
IS
IHOLD=0
YES
NO
LLL = I
10
RPMP = PMPR(LLL, IC)
R = (RPMP - 1.DO)/[(DABS(RPMP-1.DO)]
CHY1 = (RPMP ** RPMP) * (RPMP-1.0) * CHY
IS
RPMP>
1.001
YES
CHY1=(RPMP**RPMP)*CHY*R
NO
RETURN

3. PROGRAM LISTING

The program listing follows this page. The main program is listed first and each subroutine is listed in the order they are described in the PROGRAM DESCRIPTION.

	Pages
Main Program	49-54
Subroutine ASIDE	54
Subroutine VMPOUT and COSOUT	55-56
Subroutine CALOG	57
Subroutine PTIRX	58
Subroutine CALY	59
Subroutine CALVMP	60
Subroutine COSMIN	61
Subroutine ZERION	62

MATN PROGRAM

```
1        REAL*8 LEA,L10A,AREAL,Y,X(10),B(10),VX(10),A,YSUB(10),CON,YR,R,
        1YSUM,MRS,STOP,Y1,DABS,RMRS(10),PRATIO(10),PX(10),BSUM,SUMNOO,
        2 NUODLE(10),TFC(100),TVC(100),TC(100),AVC(100),AFC(100),ATC(100),
        3SUBY(100),COST(10),MC(100),AR(100),MR(100),MPX(10,100),VMPX(10,100
        4),PMPR(10,100),TR(100),YRANG1,YRANG2,RANGE,TC2,TC1,SUBTC,FC,PROFIT
        5(100), X1(10),CHY,CHY1
2        DIMENSION              EX(10),IX(10),DELX(10,30),DELB(10,30),DELEX(
        110,30)       ,DX(30),DB(30),DEX(30),JX(30),DELY(30),DELCHY(30),DEL
        2YL(30),IDY(30),DELA(30),DELCHX(30),DELXL(30),IDA(30),DES(20),
        3IHOLD(10),          DELPA(30),DELPY(30),DPX(30),DELPX(10,30)
3        COMMON TFC,TVC,TC,AVC,AFC,ATC,SUBY,MC,AR,MR,MPX,VMPX,PMPR,TR,PROFI
        1T,DES
4        DATA ISAFE / 100 /
5        DATA  X,       B,     PX / 10 *  0.DO,          10 * 0.DO, 10 *
        1 0.DO /
6        DATA  IDA , DELA, DELCHX, DELXL, DELY, DELCHY, DELYL     / 30 * 0,
        1 30 * 0.0 , 30 * 0.0 , 30 * 0.0 , 30 * 0.0 , 30 * 0.0 , 30 * 0.0 /
7        DATA  DELX , DELB, DELEX , DELPX / 300 * 0.0, 300 * 0.0,  300 *
        1 0.0 , 300 * 0.0 /
8        INDE = 1
9        LDE = 2
10       I = 0
11       J = 0
12       NFLAG = 6
13       RANGE = 0.0
14       IF ( RANGE .EQ. 0.0 ) RANGE = 1.0
15       LSIGN = 0
16     4 CONTINUE
17       IF(NFLAG .EQ.  6 ) WRITE(6,120)
18     120 FORMAT(1H1,20X,2H--,2X,22HPROBLEM IDENTIFICATION,2X,2H--/)
19       READ(5,101) NFLAG,(DES(N),N=1,19)
20     101 FORMAT(I2,2X,19(A4))
21       IF(NFLAG .EQ. 99) GO TO 99
22       WRITE(6,101) NFLAG,(DES(N),N=1,19)
23       IF(NFLAG .LE. 0 ) GO TO 4
24        WRITE(6,115)
25     115 FORMAT(1H ,6X,61H--  INPUT DATA  --  CARD IMAGES OF INPUT DATA  --
          1  INPUT DATA/)
26       IF(NFLAG .EQ. 1 ) LSIGN = 1
27     2 READ(5,100)IFLAG,Y,CHY,YLAST,PY,LX
28       CHY1 = CHY
29       IGROY = IFLAG
30       IF(IFLAG .GT. 9)  IGROW = IFLAG - 9
31       IDS = IFLAG
32       WRITE(6,142)
33     142 FORMAT(5HIFLAG,14X,1HY,18X,3HCHY,16X,5HYLAST,11X,2HPY,3X,2HLX)
34       WRITE(6,100)IFLAG,Y,CHY,YLAST,PY,LX
35     100 FORMAT(I2,2X,F20.4,F20.4,F20.4,F14.3,I2)
36       READ(5,150) IFLAG,A,CHX,XLAST,FC,IA
37       IGROX = IFLAG
38       WRITE(6,143)
39     143 FORMAT(5HIFLAG,14X,1HA,18X,3HCHX,16X,5HXLAST,11X,2HFC,3X,2HIA)
40       WRITE(6,150)IFLAG,A,CHX,XLAST,FC,IA
41       CALL ASIDE(A,IA,AREAL,LEA,L10A)
42       YR = Y
43       Y1 = Y
44     150 FORMAT(1X,I2,1X,F20.4,F20.4,F20.4,F14.3,I2)
45       DO 3  I = 1, 10
46       X1(I) = X(I)
47     3 CONTINUE
```

MAIN PROGRAM

```
48          I = 0
49        5 CONTINUE
50          I = I + 1
51          READ(5,102)IFLAG,X(I),B(I),EX(I),PX(I),IX(I)
52          X1(I) = X(I)
53      102 FORMAT(I2,2X,F20.4,F25.12,F15.9,F14.3,I2)
54          WRITE(6,144)I,I,I,I,I
55      144 FORMAT(1HO,5HIFLAG,13X,1HX,I2,15X,1HB,I2,14X,2HEX,I2,15X,2HPX,I2,
           12X,2HIX,I2)
56          WRITE(6,152)IFLAG,X(I),B(I),EX(I),PX(I),IX(I)
    C
57          IF( IX(I) .EQ. 1 ) LDE = I
58          IF( IX(I) .EQ. 2 ) INDE = I
59          FIELD = 1.0
60          IHOLD(I) = 0
61          IF(IFLAG  .EQ. 1 .OR. IFLAG .EQ. 3)  IHOLD(I)= 1
62          IF( IFLAG .NE. 2 .AND. IFLAG .NE. 3 ) GO TO 5
63          ISTOP = I
64           XMAX = XLAST
65          IF(X(INDE) .LT. XMAX) KDEC = 1
66          IF(X(INDE) .GT. XMAX) KDEC = -1
67          IF(Y .LT. YLAST) KDECY = 1
68          IF(Y .GT. YLAST) KDECY = -1
69       12 CONTINUE
70        9 CONTINUE
71          J = C
72       10 CONTINUE
73          J = J + 1
74          WRITE(6,146) J
75      146 FORMAT(1HO,49H    -    -    -    -    -    CHANGES FOR PERIOD  ,I2
           1,25H    -    -    -    -    -)
76       17 CONTINUE
77          CURVE = 1.0
78          READ(5,102) IFLAG,DX(J),DB(J),DEX(J),DPX(J),JX(J)
79          IF(IFLAG .EQ. 99 ) GO TO 14
80          IF(IFLAG .EQ. 20) DELY(J) = DX(J)
81          IF(IFLAG .EQ. 20) DELCHY(J) = DB(J)
82          IF(IFLAG .EQ. 20) DELYL(J) = DEX(J)
83          IF(IFLAG .EQ. 20) DELPY(J) = DPX(J)
84          IF(IFLAG .EQ. 20) IDY(J) = JX(J)
85          IF(IFLAG .EQ. 20)WRITE(6,140)J,J,J,J,J
86      140 FORMAT(1HO,5HIFLAG,9X,4HDELY,I2,12X,6HDELCHY,I2,13X,5HDELYL,I2,8X,
           15HDELPY,I2,3X,3HIDY,I2)
87          IF(IFLAG .EQ. 20)WRITE(6,152)IFLAG,DELY(J),DELCHY(J),DELYL(J),DELP
           1Y(J),IDY(J)
88      152 FORMAT(1X,I2,1X,F20.4,F25.12,F15.9,F14.3,I2)
    C
89          IF(IFLAG .EQ. 10) DELA(J) = DX(J)
90          IF(IFLAG .EQ. 10) DELCHX(J) = DB(J)
91          IF(IFLAG .EQ. 10) DELXL(J) = DEX(J)
92          IF(IFLAG .EQ. 10) DELPA(J) = DPX(J)
93          IF(IFLAG .EQ. 10) IDA(J) = JX(J)
94          IF(IFLAG .EQ. 10)WRITE(6,141)J,J,J,J,J
95      141 FORMAT(1HO,5HIFLAG,9X,4HDELA,I2,12X,6HDELCHX,I2,13X,5HDELXL,I2,8X,
           15HDELPA,I2,3X,3HIDA,I2)
96          IF(IFLAG .EQ. 10)WRITE(6,152)IFLAG,DELA(J),DELCHX(J),DELXL(J),
           1DELPA(J),IDA(J)
    C
97          IF(IFLAG .EQ. 10 .OR. IFLAG .EQ. 20) GO TO 11
98          IF(JX(J) .LE. 0 ) GO TO  15
99          I = JX(J)
```

MAIN PROGRAM

```
      C
100           DELX(I,J) = DX(J)
101           DELB(I,J) = DB(J)
102           DELEX(I,J) = DEX(J)
103           DELPX(I,J) = DPX(J)
104        15 CONTINUE
105           WRITE(6,145) J,J,J,J,J
106       145 FORMAT(1HO,5HIFLAG,12X,2HDX,I2,13X,2HDB,I2,13X,3HDEX,I2,14X,  5HDE
          1LPX,I2,2X,  2HJX,I2)
107           WRITE(6,152)IFLAG,DX(J),DB(J),DEX(J),DPX(J),JX(J)
108        11 CONTINUE
      C
109           IF( IFLAG .EQ. 2 ) GO TO 10
110           IF( IFLAG .EQ. 6 ) GO TO 14
111           GO TO 17
      C
112        14 CONTINUE
113           JSTOP =J
114           IF(IDA(J) .GT. 1 ) JSTOP = IDA(J)
115           J = 0
116            IF(LX .GE. 10 .AND. LX .LE. 19) GO TO 300
117        13 CONTINUE
118           I = 1
119           YSUM = 0.0
120        18 CONTINUE
121           IF( LX .LT. 10  .OR. LX .GT. 19) CALL CALOG(A,Y,ISTOP,I,X,B,J,DELX
          1,N,Y1,L10A,LEA,IA,LDE,AREAL)
122           IF(J .EQ. 0 ) X1(INDE) = X(INDE)
123       235 CONTINUE
124           WRITE(6,11C) (DES(N),N=1,19), Y
125       110 FORMAT(1H1,6X,19(A4),3X,15H  FOR YIELD OF,2X,F25.4)
      C
126           WRITE(6,111) (B(I),I = 1,10)
127       111 FORMAT(1HO,20X,25HCOEFFICIENTS OF EQUATIONS/ 5(1X,F25.12)/5(1X,F25
          1.12) )
      C
128           WRITE(6,116) AREAL,L10A,LEA
129       116 FORMAT(1HO,2X,11H - A - IS  ,E20.10,5X, 5H- * -, 5X,14H - L10A - I
          1S  ,F21.4,5X, 5H- * -, 5X,13H - LEA - IS  ,F21.4)
      C
130           WRITE(6,113) INDE,LDE
131       113 FORMAT(1HO, 4X,2HX1,11X,2HX2,11X,2HX3,11X,2HX4,          11X,2HX5,1
          11X,2HX6,11X,2HX7,11X,2HX8,11X,2HX9,11X,5HMRS X,I2,2H X,I2)
132        30 CONTINUE
133           MRS = (B(INDE) * X(LDE) )/(B(LDE) * X(INDE) )
134           WRITE(6,114) (X(I),I = 1,9),MRS
135       114 FORMAT(1X,10(E13.6))
136           IF(CHX .LE. 1.00) X(INDE) = X(INDE) + X(INDE)*CHX
137           IF(CHX .GT. 1.00) X(INDE) = X(INDE) + CHX
      C
138           IF(KDEC .EQ. 1   .AND. X(INDE) .GT. XLAST) GO TO 60
139           IF(KDEC .EQ. -1  .AND. X(INDE) .LT. XLAST) GO TO 60
      C
140           CALL CALOG(A,Y,ISTOP,I,X,B,J,DELX,N,Y1,L10A,LEA,IA,LDE,AREAL)
141           GO TO 30
142        60 CONTINUE
143           IF(CHY .LE. 1.00) Y = Y + Y*CHY
144           IF(CHY .GT. 1.00) Y = Y + CHY
145           IF(Y .GT. YLAST .AND. LX .GT. 19) GO TO 70
146           IF(Y .GT. YLAST) GO TO 300
147           X(INDE) = X1(INDE)
```

MAIN PROGRAM

```
148         FIELD  = FIELD + 1.0
      C
149         GO TO 13
150       7C CONTINUE
151         IF(IFLAG .EQ. 99 ) GO TO 99
152         J = J + 1
153         WRITE(6,2001) J
154      2001 FORMAT(1H1,5X,38H- $ - THIS IS THE BEGINNING OF PERIOD ,I2,
           16H - $ -  )
155         IF ( J .GE. JSTOP ) NFLAG = 6
156         IF( J .CE. JSTOP ) GO TO 4
157         FIELD = 1.C
158         Y1 = Y1 + DELY(J)
159         Y = Y1
160         YR = Y1
161         CHY=CHY+DELCHY(J)
162         YLAST = YLAST + DELYL(J)
163         IF(J .LE. 1 ) GO TO 75
164         IF(IDA(1) .GT. 1 ) IDA(J) = IDA(J - 1 )
165         IF(IDA(J) .GT. 1 .AND. J .GT. 1 ) DELA(J) = DELA(J - 1 )
166       75 CONTINUE
167         IF(IGROY .EQ. 2 ) A = A + A*DELA(J)
168         IF ( IGROY .NE. 2 ) A = A +  DELA(J)
169         IF(IDA(J) .GE. 1 ) A = A + A*(DELA(J) )
170         CHX = CHX + DELCHX(J)
171         XLAST = XLAST + DELXL(J)
172         DO 80  I=1,10
173         IF(IDA(1) .LE. 1 .OR. J .LE. 1 ) GO TO 77
174           IF(J .GT. 1 ) DELPX(I,J) = DELPX(I,J-1)
175           IF(J .GT. 1 ) DELEX(I,J) = DELEX(I,J-1)
176           IF(J .GT. 1 ) DELB(I,J) = DELB(I,J-1)
177       77 CONTINUE
178         IF(DELX(I,J) .GT. 1.0) X(I) = X1(I) + DELX(I,J)
179         IF(DELX(I,J) .LE. 1.0) X(I) = X1(I) + DELX(I,J) * X1(I)
180         X1(I) = X(I)
181         B(I) =B(I) + DELB(I,J)
182         EX(I)=EX(I)+ DELEX(I,J)
183         PX(I) = PX(I) + DELPX(I,J)
184         IF(IDA(J) .GT. 1  .AND. J .GT. 1 ) DELB(I,J) = DELB(I,J-1)
185         IF(IDA(J) .GE. 1) B(I) = B(I) + B(I) * DELB(I,J)
186         GO TO 78
187       78 CONTINUE
188       80 CONTINUE
189         CALL ASIDE(A,IA,AREAL,LEA,L10A)
190         IF(LX .GE. 10 .AND. LX .LT. 20 ) GO TO 300
191         GO TO 13
192      300 CONTINUE
193         Y = Y1
194         X(INDE)= X1(INDE)
195         WRITE(6,112) (DES(N),N=1,19)
196      112 FORMAT(1H1,2X,43HEXPANSION PATH - PHYSICAL INPUT & OUTPUT - ,?X,
           119(A4) )
197         WRITE(6,111) (B(I),I=1,10)
198         WRITE(6,117) (PX(I),I=1,10)
199      117 FORMAT(1HO,20X,16HPRICES OF INPUTS/5(11X,F15.3)/5(11X,F15.3))
200         WRITE(6,116) AREAL,L10A,LEA
201         IF(YR .NE. Y) WRITE(6,133)
202      133 FORMAT(1HO,34HTHE VALUE OF Y IS TO BE DETERMINED)
203         WRITE(6,118)
204      118 FORMAT(1HO,4X,2HX1,11X,2HX2,11X,2HX3,11X,2HX4,11X,2HX5,11X,2HX6,11
           1X,2HX7,11X,2HX8,11X,2HX9,10X,12HOUTPUT - Y - - )
```

MAIN PROGRAM

```
205      305 CONTINUE
206          IC = 0
207          Y = YR
208      251 CONTINUE
209          IC = IC + 1
210          YRANG1 = Y - RANGE
211          YRANG2 = Y + RANGE
212          CALL COSMIN(YRANG1 ,X,B,PX,AREAL,TC1,  INDE,ISTOP,IHOLD,IC,FC)
213          CALL COSMIN(YRANG2 ,X,B,PX,AREAL,TC2,INDE,ISTOP,IHOLD,IC,FC)
214          CALL COSMIN(Y,X,B,PX,AREAL,SUBTC,INDE,ISTOP,IHOLD,IC,FC)
215          TC(IC) = SUBTC
216          CALL CALY(Y,B,X,I,ISTOP,AREAL)
217          WRITE(6,114) (X(I),I=1,9),  Y
218          MC(IC) = (TC2 - TC1)/(2.0 * RANGE)
219          AVC(IC) = TVC(IC)/Y
220          AFC(IC) = TFC(IC)/Y
221          ATC(IC) = TC(IC)/Y
222          AR(IC) = PY
223          SUBY(IC) = Y
224          ICC = IC
225          MR(IC) = PY
226          CALL CALVMP(AREAL,X,B,PX,ISTOP,Y,ICC)
227          ICSTOP  = IC
228          IF(IGROX .EQ. 2) A = A + CHX
229          IF(IGROX .EQ. 3) A = A + A*CHX
230          CALL ASIDE(A,IA,AREAL,LEA,L10A)
231      325 CONTINUE
232      330 CONTINUE
233          IF(IC .GE. ISAFE ) GO TO 320
234          IF(IGROY .EQ. 1) GO TO 321
235          IF(CHY .LT. 1.0) Y = Y + Y * CHY
236          IF(CHY .GE. 1.0) Y = Y + CHY
237          IF(KDECY  .EQ. 1  .AND. Y .GE. YLAST) GO TO 320
238          IF(KDECY  .EQ. -1 .AND. Y .LE. YLAST) GO TO 320
239          GO TO 251
240      321 CONTINUE
241          IF( DABS(PMPR(INDE,IC) - 1.D0) .LE. 0.001 ) GO TO 320
242          CALL ZERUIN(Y,CHY1,     CHY,ISTOP,IC,IHOLD)
243          Y = Y + CHY1
244          GO TO 251
245      320 Y = Y - CHY
246          CALL PTRIX(Y,B,AREAL,L10A,LEA,PX,ISTOP,X)
247          CALL COSOUT(ICSTOP,B,AREAL,L10A,LEA,PX,FC)
248          CALL VMPOUT(Y,PX,B,AREAL,LEA,L10A,ICC,ICSTOP)
249          GO TO 70
250       99 CONTINUE
251          STOP
252          END
```

SUBROUTINE ASIDE

```
306         SUBROUTINE ASIDE(A,IA,AREAL,LEA,L10A)
307         REAL*8 L10A,LEA,AREAL,DLOG,DLOG10,DEXP,A
308         IF(IA .LT. 10) AREAL = DEXP(A)
309         IF(IA .GT. 10) AREAL = (10.0)**(A)
310         IF(IA .EQ. 10) AREAL = A
311         LEA = DLOG(AREAL)
312         L10A =  DLOG10(AREAL)
313         RETURN
314         END
```

SUBROUTINE BLOCK DATA

```
513         BLOCK DATA
514         REAL*8 TFC(100),TVC(100),TC(100),AVC(100),AFC(100),ATC(100),SUBY(1
           100),MC(100),AR(100),MR(100),MPX(10,100),VMPX(10,100),PMPR(10,100),
           1TR(100),PROFIT(100)
515         COMMON TFC,TVC,TC,AVC,AFC,ATC,SUBY,MC,AR,MR,MPX,VMPX,PMPR,TR,PROFI
           1T,DES
516         DIMENSION DES(20)
517         DATA PMPR , MPX / 1000 * 1.D0 , 1000 * 1.D0 /
518         END
```

SUBROUTINE VMPOUT

```
391          SUBROUTINE VMPOUT(Y,PX,B,AREAL,LEA,L10A,ICC,ICSTOP)
392          REAL*8 MPX(10,100),VMPX(10,100),PMPR(10,100),Y,PX(10),B(10),AREAL,
            1LEA,L10A  ,SUBY(100),MR(100)
393          REAL*8 TFC(100),TVC(100),TC(100),AVC(100),AFC(100),ATC(100),
            1    MC(100),AR(100)                    ,TR(100),PROFIT(100)
394          DIMENSION DES(20)
395          COMMON TFC,TVC,TC,AVC,AFC,ATC,SUBY,MC,AR,MR,MPX,VMPX,PMPR,TR,PROFI
            1T,DES
396          WRITE(6,121) (DES(N),N=1,19)
397      121 FORMAT(1H1,5X,25HVMPX/PX AND MP FOR   -   ,19(A4)/5X,73HRATIO OF
            1THE VALUE OF THE MARGINAL PRODUCT TO THE PRICE OF EACH RESOURCE  )
398          WRITE(6,111) (B(I),I=1,10)
399          WRITE(6,116) AREAL,L10A,LEA
400          WRITE(6,117) (PX(I),I=1,10)
401          WRITE(6,118)
402      118 FORMAT(1HO,2X,8HVMP1/PX1,5X,8HVMP2/PX2,5X,8HVMP3/PX3,5X,
            18HVMP4/PX4,5X,8HVMP5/PX5,5X,8HVMP6/PX6,5X,8HVMP7/PX7,5X,8HVMP8/PX8
            1,5X,8HVMP9/PX9,5X,1CHY - OUTPUT  )
403          WRITE(6,120)
404      120 FORMAT(3X,4HMPX1,9X,4HMPX2,9X,4HMPX3,9X,4HMPX4,9X,4HMPX5,9X,4HMPX6
            1,9X,4HMPX7,9X,4HMPX8,9X,4HMPX9,9X,2HMR)
405          DO 10  ICC = 1,ICSTOP
406          WRITE(6,119) (PMPR(IN,ICC) , IN = 1,9) , SUBY(ICC)
407      119 FORMAT(1HO,10(E13.6) )
408          WRITE(6,199) (MPX(IN,ICC) ,IN = 1,9) , MR(ICC)
409      199 FORMAT(1X,10(E13.6) )
410       10 CONTINUE
411      111 FORMAT(1HO,20X,25HCOEFFICIENTS OF EQUATIONS/ 5(1X,F25.12)/5(1X,F25
            1.12) )
412      116 FORMAT(1HO,2X,11H - A - IS  ,E20.10,5X, 5H- * -, 5X,14H - L10A - I
            1S  ,F21.4,5X, 5H- * -, 5X,13H - LEA - IS  ,F21.4)
413      117 FORMAT(1HO,20X,16HPRICES OF INPUTS/5(11X,F15.3)/5(11X,F15.3) )
414          RETURN
415          END
```

SUBROUTINE COSOUT

```
      SUBROUTINE COSOUT(ICSTOP,B,AREAL,L10A,LEA,PX,FC)
      REAL*8 TFC(100),TVC(100),TC(100),AVC(100),AFC(100),ATC(100),SUBY(1
     100),TR(100),AR(100),MC(100),B(10),PX(10),LEA,L10A,AREAL,FC,PROFIT
     2(100)
      REAL*8 MR(100),MPX(10,100),VMPX(10,100),PMPR(10,100)
      DIMENSION DES(20)
      COMMON TFC,TVC,TC,AVC,AFC,ATC,SUBY,MC,AR,MR,MPX,VMPX,PMPR,TR,PROFI
     1T,DES
      WRITE(6,121) (DES(N), N = 1,19),FC,AR(1)
  121 FORMAT(1H1,5X,25HEXPANSION PATH FOR   -    ,19(A4)/5X,58H  - COST
     1FUNCTIONS    *    *    *  FIXED  COST  IS   -    ,F15.3,6X,
     219HFIRST  PRICE OF  Y ,F25.3 )
      WRITE(6,111) (B(I),I = 1,10)
      WRITE(6,116) AREAL,L10A,LEA
      WRITE(6,117) (PX(I),I = 1,10)
      WRITE(6,126)
  126 FORMAT(1HO,3X,1HY,12X,2HTC,10X,          3HTVC,10X,3HATC,10X,3HAFC
     1,10X,3HAVC,11X,2HMC,9X,5H  MR ,10X,2HTR,10X,6HPROFIT )
      DO 270  ICC = 1,ICSTOP
      TR(ICC) =   SUBY(ICC) * AR(ICC)
      PROFIT(ICC) = TR(ICC) - TC(ICC)
      WRITE(6,127) SUBY(ICC),TC(ICC),          TVC(ICC),ATC(ICC),AFC(ICC)
     2,AVC(ICC),MC(ICC),AR(ICC),TR(ICC),PROFIT(ICC)
  127 FORMAT(1X,10(E13.6) )
  111 FORMAT(1HO,20X,25HCOEFFICIENTS OF EQUATIONS/ 5(1X,F25.12)/5(1X,F25
     1.12) )
  116 FORMAT(1HO,2X,11H - A - IS  ,E20.10,5X, 5H- * -, 5X,14H - L10A - I
     1S  ,F21.4,5X, 5H- * -, 5X,13H - LEA - IS  ,F21.4)
  117 FORMAT(1HO,20X,16HPRICES OF INPUTS/5(11X,F15.3)/5(11X,F15.3)/)
  270 CONTINUE
      RETURN
      END
```

SUBROUTINE CALOG

```
253          SUBROUTINE CALOG(A,Y,ISTOP,I,X,B,J,DELX,N,Y1,L10A,LEA,IA,LDE,AREAL
            1)
254          REAL*8 L10A,LEA,DLOG,DLOG10,     Y,AREAL,DEXP,XLLDE,X(10),B(10),
            1VX(10),A,XSUB(10),DABS,ASUB,YSUM,Y1
255           DIMENSION DELX(10,30)
256          IF(IA .EQ. 10) GO TO 205
257          ASUB = DABS(A)
258          IF(IA .LT. 10) YSUM = LEA
259          IF(IA .GT. 10) YSUM = L10A
260          DO 230  I = 1, ISTOP
261          IF(I .EQ. LDE .AND. Y .NE. 0.0) GO TO 227
262          XSUB(I) = DABS(X(I))
263          IF(X(I) .GT. 0.0 .AND. IA .LT. 10) VX(I) = B(I) * DLOG(XSUB(I))
264          IF(X(I) .LT. 0.0 .AND. IA .LT. 10) VX(I) = -B(I) * DLOG(XSUB(I))
265          IF(X(I) .GT. 0.0 .AND. IA .GT. 10) VX(I) = B(I) * DLOG10(XSUB(I))
266          IF(X(I) .LT. 0.0 .AND. IA .GT. 10) VX(I) = -B(I) * DLOG10(XSUB(I))
267          IF(X(I) .EQ. 0.0) GO TO 227
268          YSUM = YSUM + VX(I)
269      227 CONTINUE
270      230 CONTINUE
271          IF(Y .EQ. 0.0) GO TO 240
272          IF(IA .LT. 10) XLLDE = (DLOG(Y)-YSUM)/B(LDE)
273          IF(IA  .GT. 10) XLLDE =(DLOG10(Y) - YSUM)/B(LDE)
274          IF(IA .LT. 10) X(LDE) =DEXP(XLLDE)
275          IF(IA .GT. 10) X(LDE) = (10.0)**(XLLDE)
276      240 CONTINUE
         C
277          IF(Y .EQ. 0.0 .AND. IA .LT. 10) Y =DEXP(YSUM)
278          IF(Y .EQ. 0.0 .AND. IA .GT. 10) Y = (10.0)**(YSUM)
279          RETURN
280      205 CONTINUE
281          YSUM = AREAL
282          DO 235 I = 1,ISTOP
283          IF(I .EQ. LDE) GO TO 237
284          IF(X(I) .EQ. 0.0) X(I) = 1.0
285          IF( B(I) ) 210, 215, 220
286      210 VX(I) = 1.0/(X(I)**DABS(B(I)))
287          GO TO 225
288      215 VX(I) = 1.0
289          GO TO 225
290      220 VX(I) = X(I) * * B(I)
291      225 CONTINUE
292          IF(X(I) .EQ. 0.0) VX(I) = 1.0
293          YSUM = YSUM * VX(I)
294      237 CONTINUE
295      235 CONTINUE
296          IF(Y .NE. 0.0) VX(LDE) = Y / YSUM
297          IF(Y .EQ. 0.0) Y = YSUM
298          IF(B(LDE) .EQ. 0.0) X(LDE) = 1.0
299          IF(B(LDE) .EQ. 0.0) GO TO 245
300          IF(B(LDE) .LT. 0.0) X(LDE) =  1.0/(VX(LDE) * * (1.0/ABS(B(LDE))))
301          IF(B(LDE) .LE. 0.0) GO TO 245
302          IF(B(LDE) .GT. 0.0) X(LDE) = VX(LDE) * * (1.0/B(LDE))
303      245 CONTINUE
304          RETURN
305          END
```

SUBROUTINE PTRIX

```
315         SUBROUTINE PTRIX(Y,B,AREAL,L10A,LEA,PX,ISTOP,X)
316         REAL*8 L10A,LEA,AREAL,Y,X(10),B(10),RMRS(10),PRATIO(10),PX(10)
317         DIMENSION DES(20)
318         REAL*8 TFC(100),TVC(100),TC(100),AVC(100),AFC(100),ATC(100),SUBY(1
           100),MC(100),AR(100),MR(100),MPX(10,100),VMPX(10,100),PMPR(10,100),
           1TR(100),PROFIT(100)
319         COMMON TFC,TVC,TC,AVC,AFC,ATC,SUBY,MC,AR,MR,MPX,VMPX,PMPR,TR,PROFI
           1T,DES
320         WRITE(6,121) (DES(N),N=1,19),Y
321       121 FORMAT(1H1,33HMRS - PRICE RATIO MATRIX FOR  -   ,19A4/77H  - MARGIN
           1AL RATE OF SUBSTITUTION AND PRICE RATIOS  -    FOR OUTPUT LEVEL OF
           2,2X,F25.4)
322         WRITE(6,111) (B(I),I=1,10)
323         WRITE(6,116) AREAL,L10A,LEA
324         WRITE(6,117) (PX(I),I=1,10)
325       122 FORMAT(9X,8HMRS(X1,X,I1,1H),2X,9H MRS(X2,X,I1,1H),2X,9H MRS(X3,X,I
           11,1H),2X,9H MRS(X4,X,I1,1H),2X,9H MRS(X5,X,I1,1H),3X,8HMRS(X6,X,I1
           2,1H),3X,8HMRS(X7,X,I1,1H),2X,9H MRS(X8,X,I1,1H),2X,9H MRS(X9,X,I1,
           31H)  )
326       124 FORMAT(9X,2HPX,I2,4H/PX1,5X,2HPX,I2,4H/PX2 ,5X,2HPX,I2,4H/PX3,5X,2H
           1PX,I2,4H/PX4,5X,2HPX,I2,4H/PX5,5X,2HPX,I2, 4HPX6,5X,2HPX,I2,
           2 4H/PX7,5X,2HPX,I2,4H/PX8,5X,2HPX,I2,4H/PX9)
327         DO 268  I=1,10
328         RMRS(I) = 0.0
329         PRATIO(I)= 0.0
330       268 CONTINUE
331         DO 270  INOTI =1,ISTOP
332         I2 = INOTI
333         WRITE(6,122) I2,I2,I2,I2,I2,I2,I2,I2,I2
334         WRITE(6,124) I2,I2,I2,I2,I2,I2,I2,I2,I2
335         DO 275  I = 1,ISTOP
336         IF(B(I) .EQ. 0.0 .OR. X(INOTI) .EQ. 0.0 .OR. PX(I) .EQ. 0.0 )
           1 GO TO 273
337         RMRS(I) = (B(INOTI) * X(I))/(B(I) * X(INOTI))
338         PRATIO(I) = PX(INOTI)/PX(I)
339       273 CONTINUE
340       275 CONTINUE
341         WRITE(6,123) (RMRS(I), I = 1,9), INOTI,(PRATIO(I),I=1,9),
           1INOTI
342       123 FORMAT(8X,9(E13.6)/2X,1HX,I1,4X,9(E13.6),3X,1HX,I1 / )
343       111 FORMAT(1HO,20X,25HCOEFFICIENTS OF EQUATIONS/ 5(1X,F25.12)/5(1X,F25
           1.12) )
344       116 FORMAT(1HO,2X,11H - A - IS  ,E20.10,5X, 5H- * -, 5X,14H - L10A - I
           1S  ,F21.4,5X, 5H- * -, 5X,13H - LEA - IS  ,F21.4)
345       117 FORMAT(1HO,20X,16HPRICES OF INPUTS/5(11X,F15.3)/5(11X,F15.3)/)
346       270 CONTINUE
347         RETURN
348         END
```

SUBROUTINE CALY

```
      SUBROUTINE CALY(Y,B,X,I,ISTOP,AREAL)
      REAL*8 AREAL,Y,X(10),B(10),VX(10),YSUM,DABS
      YSUM = AREAL
      DO 235 I = 1,ISTOP
      IF(X(I) .EQ. 0.0) X(I) = 1.0
      IF( B(I) ) 210, 215, 220
  210 VX(I) = 1.0/(X(I)**DABS(B(I)))
      GO TO 225
  215 VX(I) = 1.0
      GO TO 225
  220 VX(I) = X(I) * * B(I)
  225 CONTINUE
      IF(X(I) .EQ. 0.0) VX(I) = 1.0
      YSUM = YSUM * VX(I)
  237 CONTINUE
  235 CONTINUE
       Y = YSUM
      RETURN
      END
```

SUBROUTINE CALVMP

```
      SUBROUTINE CALVMP(AREAL,X,B,PX,ISTOP,Y,ICC)
      REAL*8 X(10),B(10),PX(10),MPX(10,100),VMPX(10,100),PMPR(10,100),Y,
     1MR(100),VX(10) , DABS,AREAL,SB
      REAL*8 TFC(100),TVC(100),TC(100),AVC(100),AFC(100),ATC(100),SUBY(1
     100),MC(100),AR(100),TR(100),PROFIT(100)
      DIMENSION DES(20)
      COMMON TFC,TVC,TC,AVC,AFC,ATC,SUBY,MC,AR,MR,MPX,VMPX,PMPR,TR,PROFI
     1T,DES
      DO 10 INOTI = 1,ISTOP
      PMPR(INOTI,ICC) = 0.0
      MPX(INOTI,ICC) = AREAL
      DO 20 I = 1, ISTOP
      IF(X(I) .EQ. 0.0) X(I) = 1.0
      IF(I .EQ. INOTI) GO TO 19
      IF( B(I) ) 210,215,220
  210 VX(I) = 1.0/(X(I) ** DABS(B(I)) )
      GO TO  226
  215 VX(1.0) = 1.0
      GO TO 226
  220 VX(I) = X(I) ** B(I)
      GO TO 226
   19 CONTINUE
      SB = (B(I) - 1.0)
      IF (SB) 211,216,221
  211 VX(I) = B(I)*(1.0/(X(I) ** DABS(SB)))
      GO TO 226
  216 VX(I) = B(I)
      GO TO 226
  221 CONTINUE
      VX(I) = B(INOTI) * X(I) ** SB
  226 CONTINUE
      MPX(INOTI,ICC) = MPX(INOTI,ICC) * VX(I)
   20 CONTINUE
      VMPX(INOTI,ICC) = MR(ICC) * MPX(INOTI,ICC)
      PMPR(INOTI,ICC) = VMPX(INOTI,ICC) / PX(INOTI)
   10 CONTINUE
      RETURN
      END
```

SUBROUTINE COSMIN

```
451         SUBROUTINE COSMIN(Y,X,B,PX,AREAL,SUBTC,INDE,ISTOP,IHOLD,IC,FC)
452         REAL*8 Y,X(10),B(10),PX(10),AREAL,TFC(100),TVC(100), BSUM,
           1SUMNOO,NOODLE(10),DABS,A1,CON,COST(10),SUBTC,TC1,TC2,YRANG1,
           2YRANG2,FC
453         REAL*8                                  AVC(100),AFC(100),ATC(100),SUBY(1
           100),MC(100),AR(100),MR(100),MPX(10,100),VMPX(10,100),PMPR(10,100),
           1TR(100),PROFIT(100),TC(100)
454           DIMENSION  IHOLD(10),DES(20)
455         COMMON TFC,TVC,TC,AVC,AFC,ATC,SUBY,MC,AR,MR,MPX,VMPX,PMPR,TR,PROFI
           1T,DES
456         NDE = INDE
457         IF( IHOLD(INDE) .EQ. 0 ) GO TO 5
458         DO 10  I = 1,     ISTOP
459         IF( IHOLD(I) .EQ. 0 ) INDE = I
460         IF( IHOLD(I) .EQ. 0 ) GO TO 5
461      10 CONTINUE
462       5 CONTINUE
463         BSUM = 0.0
464         SUMNOO  = 1.0
465         DO 257 I=1,ISTOP
466         IF( B(I) .EQ. 0.0) GO TO 256
467         IF(IHOLD(I) .EQ. 1) GO TO 256
468         BSUM = BSUM + B(I)
469         NOODLE(I) = ((PX(INDE) * B(I)) / (B(INDE) * PX(I)))**B(I)
470         SUMNOO = SUMNOO * NOODLE(I)
471      256 CONTINUE
472      257 CONTINUE
473         A1 = DABS(AREAL)
474         DO 220 I = 1,ISTOP
475         IF(IHOLD(I) .EQ. 1) A1 =  A1 * ( X(I) ** B(I))
476      220 CONTINUE
477         X(INDE) = ((Y/A1)/SUMNOO)**(1.0/BSUM)
478          CON = (X(INDE)*PX(INDE))/B(INDE)
479         DO 250  I = 1,   ISTOP
480         IF(IHOLD(I) .EQ. 1) GO TO 255
481         X(I) = CON * (B(I)/PX(I))
482      255 CONTINUE
483      250 CONTINUE
484      260 CONTINUE
485      265 CONTINUE
486         TFC( IC) = FC
487         TVC(IC)  = 0.0
488      310 CONTINUE
489         DO 320  I =1,ISTOP
490         COST(I) = X(I)*PX(I)
491         IF(B(I) .EQ. 0.0 ) COST(I) = 0.0
492         IF(IHOLD(I) .EQ. 1) TFC(IC) = TFC(IC) + COST(I)
493         IF(IHOLD(I) .EQ. 0 ) TVC(IC) = TVC(IC) + COST(I)
494      320 CONTINUE
495         SUBTC = TFC(IC) + TVC(IC)
496         INDE = NDE
497         RETURN
498         END
```

SUBROUTINE ZEROIN

```
499          SUBROUTINE ZERCIN(Y,CHY1,CHY,ISTOP,IC,IHOLD)
500          REAL*8   Y,PMPR(10,100),CHY,CHY1,RPMP,DABS,R
501          REAL*8 TFC(100),TVC(100),TC(100),AVC(100),AFC(100),ATC(100),SUBY(1
.           100),MC(100),AR(100),MR(100),MPX(10,100),VMPX(10,100),
            1TR(100),PROFIT(100)
502          DIMENSION IHOLD(10),DES(20)
503          COMMON TFC,TVC,TC,AVC,AFC,ATC,SUBY,MC,AR,MR,MPX,VMPX,PMPR,TR,PROFI
            1T,DES
504          DO 10 I = 1 , ISTOP
505          IF(IHOLD(I) .EQ. 0 ) LLL = I
506       10 CONTINUE
507          RPMP = PMPR(LLL,IC)
508          R = (RPMP - 1.DO ) / (DABS(RPMP - 1.DO ))
509          CHY1 =(RPMP ** RPMP) * (RPMP - 1.0) * CHY
510          IF(RPMP .GT. 1.001) CHY1 = (RPMP ** RPMP ) * CHY * R
511          RETURN
512          END
```

4. SAMPLE PROGRAM RUN

A sample program run will be used to illustrate the printout for this program. The printout is divided into several parts. The first part is a printout of the input file including sample problem identification cards, the description of the input data and a printout of the data card images. This part will be described below under "Input File." The rest of the printout is described under "Output File." The output file can be run with two options which consist of the expansion path and/or the isoquants. The expansion path has four tables: (1) PHYSICAL INPUT AND OUTPUT, (2) MRTS-PRICE RATIO MATRIX, (3) COST FUNCTION, (4) VMPX/PX AND MP (ratio of the value of the Marginal Product to the Resource Price and Marginal Product). Each of these will be described below under its own heading. The format for the isoquant tables are all like the PHYSICAL INPUT AND OUTPUT table except the word ISOQUANT is substituted for the word "EXPANSION PATH" and "FOR OUTPUT LEVEL OF" is printed at the end of the first line. Thus no example of one is given. The slight differences between these two tables wil' be explained in the section dealing with "PHYSICAL INPUT AND OUTPUT."

Input File/Sample Data

The following page is an example of the printout of the input file. The first part, labeled "---PROBLEM IDENTIFICATION---", contains the printout of the description cards. The zero's are caused by the computer reading blanks in the first two spaces on each description card except the last, which contains the number 1. This last card's contents will be printed at the top of each table in the output file. Next comes the original set of parameters labeled "---INPUT DATA--CARD IMAGES OF INPUT DATA--INPUT DATA."

The Y card shows that the original value of Y is 300.0, Y will increase by 10.0 (CHY=10.0) until Y=400 (YLAST=400), the price of Y, (PY), is 20 and since LX=11, only the expansion path will be calculated and printed. The A card shows that A=6.66 and since IA=10, then A=AREAL=6.66. Fixed Cost=0.0. If an isoquant were to be printed, X(INDE) would change by 10.0 (CHY=10.0) until it reached 400 (XLAST=400).

In this example, the user had decided to have nine resources (X1 to X9), each has a coefficient and a price. For example, the coefficient of X4 is 0.06 and the price (PX4) is 6.00. Since IX1 is 2, X1 is X(inde) and since IX2 is 1, X(LDE) is X2. The reader should note that IFLAG for X6 is 1 and therefore X6 will be held constant. IFLAG for X9 is 2, which signals the computer that this is the last resource.

The next set of input data is changes in variable values. This example has no changes but the card with a 6 in column two is read as a change card. Thus the program prints out the change heading, "--CHANGES FOR PERIOD 1--", and the heading for the change card values. Each change period begins with this change heading.

-- PROBLEM IDENTIFICATION --

0 TEST DATA FOR FACTOR - FACTOR II , TO BE USED TO EXERCISE THE PROGRAM
0 DATA FROM DR. BILLINGSLEY FOR TESTING THE MODEL CLASS ROOM USAGE
1 TEST DATA - CLASS ROOM USAGE FROM DR. BILLINGSLEY

-- INPUT DATA -- CARD IMAGES OF INPUT DATA -- INPUT DATA

FLAG	Y	CHY	YLAST	PY	LX
0	30C.0000	1C.0000	40C.0000	20.000	11
FLAG	**A**	**CHX**	**XLAST**	**FC**	**IA**
0	6.660C	1C.0000	40C.0000	0.000	10

IFLAG	X 1	B 1	EX 1	PX 1	IX 1
0	10C.0000	0.3CC00CCC0000	0.00000000	9.000	0

IFLAG	X 2	B 2	EX 2	PX 2	IX 2
0	10C.0000	0.2C0000CC0C0C	0.00000000	8.000	1

IFLAG	X 3	B 3	EX 3	PX 3	IX 3
0	10C.0000	0.1CC000C0000C	0.00000000	7.000	2

IFLAG	X 4	B 4	EX 4	PX 4	IX 4
0	10C.0000	C.C6CCCCC00000	0.00000000	6.000	0

IFLAG	X 5	B 5	EX 5	PX 5	IX 5
0	10C.0000	0.C535CCCCC0CC	0.00000000	5.500	0

IFLAG	X 6	B 6	EX 6	PX 6	IX 6
1	10C.0000	0.C350C0CC0C0C	0.00000000	4.500	0

IFLAG	X 7	B 7	EX 7	PX 7	IX 7
0	10C.0000	0.C250C0CC0CC0	0.00000000	3.500	0

IFLAG	X 8	B 8	EX 8	PX 8	IX 8
0	10C.0000	0.C2C0CC000000	0.00000000	2.500	0

IFLAG	X 9	B 9	EX 9	PX 9	IX 9
2	10C.0000	C.C1CC00CC000C	0.00000000	1.500	0

- - - - - CHANGES FOR PERIOD 1 - - - - -

IFLAG	DX 1	DB 1	DEX 1	DELPX 1	JX 1
6	C.0000	0.CCCCCCCC 00	C.00000000	0.000	0

OUTPUT FILE

(1) Physical Input and Output.

The following page is an example of the printout of the EXPANSION PATH - PHYSICAL INPUT AND OUTPUT. Along with this label, the contents of the last description card make up the first line of the printout. The next line prints COEFFICIENTS OF EQUATIONS followed by two lines of B values. The first five coefficients are printed in the first line and the last four are printed in the second line, (B1=0.3, B2=0.2, B3=0.1, B4=0.6, B5=0.535, etc.). Next comes the resource prices labeled PRICES OF INPUTS, (P1=9.00, P2=8.00, P3=7.00, P4=6.00, P5=5.5, etc.). This is followed by the A values (AREAL=6.66, L10A=0.8235, LEA=1.8961).

Finally, the actual physical values of the expansion path are printed. The values are double precision, exponential form. For an output, (Y), level of 300, X1 is 184.563, X2 is 138.422 and so forth. Output goes from 300 to 400 by increments of 10 and the cost minimizing level of each resource increases with each level of output. For example, the minimum cost quantity of X3 is 55.3689 at an output of 300 and 80.5083 at an output level of 400. The quantity of X6 however, remains at 100 because this resource is being held constant, (See description of Input File). Since X6 is constant in quantity this table shows a partial expansion path.

If this were an isoquant table the first line would have the word ISOQUANT instead of EXPANSION PATH. The first line would also have the phrase, "FOR OUTPUT LEVEL OF," followed by the output level. The columns of resource quantities would be similar except that each resource would be held constant (in our example at 100) except for X(INDE) and X(LDE). X(INDE) would be incremented by CHX and X(LDE) would be calculated.

EXPANSION PATH - PHYSICAL INPUT & OUTPUT - TEST DATA - CLASS ROOM USAGE FROM DR. BILLINGSLEY

COEFFICIENTS OF EQUATIONS

0.300000C00000	C.200000000000	0.100000000000	0.060000000000	0.053500000000
0.035000000000	C.025000000000	0.020000000000	0.010000000000	0.000000000000

PRICES OF INPUTS

9.000	8.000	7.000	6.000	5.500
4.500	3.500	2.500	1.500	0.000

- A - IS 0.66600000000 01 - * - - L10A - IS 0.8235 - * - - LEA - IS 1.8961

X1	X2	X3	X4	X5	X6	X7	X8	X9	OUTPUT - Y -
0.184563D 03	0.138422D 03	C.790985D C2	0.553689D 02	0.538589D 02	0.100000D 03	0.395492D 02	0.442951D 02	0.369126D 02	0.300000C 03
0.192608D 03	0.144456D 03	0.825464D C2	0.577825D 02	0.562066D 02	0.100000D 03	0.412732D 02	0.462260D 02	0.385217D 02	0.310000D 03
0.200732D C3	0.150549D 03	0.860280D C2	0.602196D 02	0.585773D 02	0.100000D 03	0.430140D 02	0.481757D C2	0.401464D 02	0.320000C 03
0.208933D 03	0.1567000 03	0.895426D C2	0.626798D 02	0.609704D 02	0.100000D 03	0.447713D 02	0.501439D 02	0.417865D 02	0.330000D 03
0.217209D C3	0.162906D 03	0.930894D C2	0.651626D 02	0.633854D 02	0.100000D 03	0.465447D 02	0.521301D 02	0.434417D 02	0.340000C 03
0.225558D C3	0.169169D 03	0.966678D C2	0.676674D 02	0.658220D 02	0.100000D 03	0.483339D 02	0.541339D 02	0.4511160 02	0.350000D 03
0.233980D 03	0.175485D 03	0.100277D 03	0.701939D 02	0.6827960 02	0.100000D 03	0.501385D 02	0.561551D 02	0.467960D 02	0.360000C 03
0.242472D 03	0.181854D 03	0.103917D C3	0.727417D 02	0.707578D 02	0.100000D 03	0.519583D 02	C.581933D 02	0.484944D 02	0.370000D 03
0.251034D 03	0.188276D 03	0.107586D 03	0.753102D 02	0.732563D 02	0.100000D 03	0.537930D 02	0.602482D 02	0.502068D 02	0.380000C 03
C.259664D C3	0.194748D 03	0.111285D 03	0.778993D 02	0.757747D 02	0.100000D 03	0.556423D 02	0.623194D 02	0.519328D 02	0.390000D 03
0.268361D 03	0.201271D 03	0.115012D C3	0.805083D 02	0.783127D 02	0.100000D 03	0.575060D 02	C.644067D 02	0.536722D 02	0.400000C 03

(2) MRTS - PRICE RATION MATRIX

The following page is an example of the printout of the MRS - PRICE RATIO MATRIX. The first few lines are similar to those of the PHYSICAL INPUT AND OUTPUT TABLE. This table prints out the output level at the time the subroutine PTRIX was called. In this case, the output level is 410. This is followed by the COEFFICIENTS OF EQUATIONS. (The B values), the values for AREAL, L1OA, and LEA and the PRICES OF THE INPUTS.

Then comes the actual Matrix. It is a nine by nine matrix and each element is actually a compound element in that it is made up of four parts. The top two parts are the labels and the bottom two parts are the numerical values corresponding to these labels. The first label, MRTS (X /X) corresponds to the first numerical value (the third part of the element) and the second label (PX /PX) corresponds to the second numerical value (the fourth part of the element). Consider for example the following two elements.

Row 5 Column 3	Row 4 Column 6
MRTS(X3,X5)	MRTS(X6,X4)
PX5/PX3	PX4/PX6
0.785714D 00	0.212933D 01
0.785714D 00	0.133333D 01

Row 5, Column 3 says that the MRTS between resource 3 and resource 5 is 0.785714. Since the price ratio is also the same, these resources are at their cost minimization level. Consider now resource 6 which is being held constant. The MRTS between resource 6 and resource 4 is 2.12933 and the price ratio is 1.33333, (see Column 6, Row 4). Thus resource six is not at its cost minimization level. A similar condition exists for all elements in the sixth row and the sixth column. But X4 is at its cost minimization level as an inspection of the other elements in Column 4 and Row 4 will verify.

MRS - PRICE RATIO MATRIX FOR - TEST DATA - CLASS ROOM USAGE FROM DR. BILLINGSLEY
- MARGINAL RATE OF SUBSTITUTION AND PRICE RATIOS - FOR OUTPUT LEVEL OF , 40C.0000

COEFFICIENTS OF EQUATIONS

0.3000000C0000	C.2000000000000	0.1C0000000000	0.C60C00C00C00	0.C53500000000
0.035000C00000	0.025000000000	0.020000000000	0.01C000000000	0.000000000000

- A - IS 0.666C0C00000 01 - * - - L10A - IS 0.8235 - * - - LEA - IS 1.8961

PRICES OF INPUTS

9.000	8.000	7.000	6.000	5.500
4.500	3.500	2.500	1.500	0.000

	MRS(X1,X1) PX 1/PX1	MRS(X2,X1) PX 1/PX2	MRS(X3,X1) PX 1/PX3	MRS(X4,X1) PX 1/PX4	MRS(X5,X1) PX 1/PX5	MRS(X6,X1) PX 1PX6,	MRS(X7,X1) PX 1/PX7	MRS(X8,X1) PX 1/PX8	MRS(X9,X1) PX 1/PX9	
	C.1000000 01	0.1125000 01	0.1285710 01	0.1500000 01	0.1636360 01	0.3193990 01	0.2571430 01	C.3600000 01	0.6000000 01	
X1	0.1000000 01	0.1125000 01	0.1285710 01	0.1500000 01	0.1636360 01	0.2000000 01	0.2571430 01	C.3600000 01	0.6000000 01	X1
	MRS(X1,X2) PX 2/PX1	MRS(X2,X2) PX 2/PX2	MRS(X3,X2) PX 2/PX3	MRS(X4,X2) PX 2/PX4	MRS(X5,X2) PX 2/PX5	MRS(X6,X2) PX 2PX6,	MRS(X7,X2) PX 2/PX7	MRS(X8,X2) PX 2/PX8	MRS(X9,X2) PX 2/PX9	
	0.888889D 00	0.1000000 01	0.1142860 01	C.1333330 01	0.1454550 01	0.2839100 C1	0.2285710 01	0.3200000 C1	0.5333330 01	
X2	0.888889D 00	0.1000000 01	0.1142860 01	0.1333330 01	0.1454550 01	0.1777780 01	0.2285710 01	C.3200000 01	0.5333330 01	X2
	MRS(X1,X3) PX 3/PX1	MRS(X2,X3) PX 3/PX2	MRS(X3,X3) PX 3/PX3	MRS(X4,X3) PX 3/PX4	MRS(X5,X3) PX 3/PX5	MRS(X6,X3) PX 3PX6,	MRS(X7,X3) PX 3/PX7	MRS(X8,X3) PX 3/PX8	MRS(X9,X3) PX 3/PX9	
	0.777778D 00	0.8750000 00	0.1000000 01	0.1166670 01	0.1272730 01	0.2484210 01	0.2000000 01	C.2800000 01	0.4666670 01	
X3	0.777778D 00	0.8750000 00	0.1000000 01	0.1166670 01	0.1272730 01	0.1555560 01	0.2000000 01	C.2800000 01	0.4666670 01	X3
	MRS(X1,X4) PX 4/PX1	MRS(X2,X4) PX 4/PX2	MRS(X3,X4) PX 4/PX3	MRS(X4,X4) PX 4/PX4	MRS(X5,X4) PX 4/PX5	MRS(X6,X4) PX 4PX6,	MRS(X7,X4) PX 4/PX7	MRS(X8,X4) PX 4/PX8	MRS(X9,X4) PX 4/PX9	
	0.6666670 00	0.7500000 00	0.8571430 00	0.1000000 01	0.1090910 01	0.2129330 01	0.1714290 01	0.2400000 01	0.4000000 01	
X4	0.6666670 00	0.7500000 00	0.8571430 00	0.1000000 01	0.1090910 01	0.1333330 01	0.1714290 01	C.2400000 01	0.4000000 01	X4
	MRS(X1,X5) PX 5/PX1	MRS(X2,X5) PX 5/PX2	MRS(X3,X5) PX 5/PX3	MRS(X4,X5) PX 5/PX4	MRS(X5,X5) PX 5/PX5	MRS(X6,X5) PX 5PX6,	MRS(X7,X5) PX 5/PX7	MRS(X8,X5) PX 5/PX8	MRS(X9,X5) PX 5/PX9	
	0.6111110 00	0.6875000 00	0.7857140 00	0.9166670 00	0.1000000 01	0.1951880 01	0.1571430 01	0.2200000 01	0.3666670 01	
X5	0.6111110 00	0.6875000 00	0.7857140 00	0.9166670 00	0.1000000 01	0.1222220 01	0.1571430 01	C.2200000 01	0.3666670 01	X5
	MRS(X1,X6) PX 6/PX1	MRS(X2,X6) PX 6/PX2	MRS(X3,X6) PX 6/PX3	MRS(X4,X6) PX 6/PX4	MRS(X5,X6) PX 6/PX5	MRS(X6,X6) PX 6PX6,	MRS(X7,X6) PX 6/PX7	MRS(X8,X6) PX 6/PX8	MRS(X9,X6) PX 6/PX9	
	0.3130880 00	0.3522240 00	0.4025420 00	C.4696320 00	0.5123260 00	0.1000000 01	0.8050830 00	0.1127120 01	0.1878530 01	
X6	0.5000000 00	0.5625000 00	0.6428570 00	C.7500000 00	0.8181820 00	0.1000000 01	0.1285710 01	0.1800000 01	0.3000000 01	X6
	MRS(X1,X7) PX 7/PX1	MRS(X2,X7) PX 7/PX2	MRS(X3,X7) PX 7/PX3	MRS(X4,X7) PX 7/PX4	MRS(X5,X7) PX 7/PX5	MRS(X6,X7) PX 7PX6,	MRS(X7,X7) PX 7/PX7	MRS(X8,X7) PX 7/PX8	MRS(X9,X7) PX 7/PX9	
	0.3888890 00	0.4375000 00	0.5000000 00	0.5833330 00	0.6363640 00	0.1242110 01	0.1000000 01	0.1400000 01	0.2333330 01	
X7	0.3888890 00	0.4375000 00	0.5000000 00	0.5833330 00	0.6363640 00	0.7777780 00	0.1000000 01	C.1400000 01	0.2333330 01	X7
	MRS(X1,X8) PX 8/PX1	MRS(X2,X8) PX 8/PX2	MRS(X3,X8) PX 8/PX3	MRS(X4,X8) PX 8/PX4	MRS(X5,X8) PX 8/PX5	MRS(X6,X8) PX 8PX6,	MRS(X7,X8) PX 8/PX7	MRS(X8,X8) PX 8/PX8	MRS(X9,X8) PX 8/PX9	
	0.2777780 00	0.3125000 00	0.3571430 00	0.4166670 00	0.4545450 00	0.8872190 00	0.7142860 00	0.1000000 01	0.1666670 01	
X8	0.2777780 00	0.3125000 00	0.3571430 00	0.4166670 00	0.4545450 00	0.5555560 00	0.7142860 00	C.1000000 01	0.1666670 01	X8
	MRS(X1,X9) PX 9/PX1	MRS(X2,X9) PX 9/PX2	MRS(X3,X9) PX 9/PX3	MRS(X4,X9) PX 9/PX4	MRS(X5,X9) PX 9/PX5	MRS(X6,X9) PX 9PX6,	MRS(X7,X9) PX 9/PX7	MRS(X8,X9) PX 9/PX8	MRS(X9,X9) PX 9/PX9	
	0.1666670 00	0.1875000 00	0.2142860 00	C.2500000 00	0.2727270 00	0.5323320 00	0.4285710 00	C.6000000 00	0.1000000 01	
X9	0.1666670 00	0.1875000 00	0.2142860 00	C.2500000 00	0.2727270 00	0.3333330 00	C.4285710 00	C.6000000 00	0.1000000 01	X9

T2

(3) COST FUNCTIONS

The following page is an example of the printout of the table of the various cost values at each level of output. The first few lines are similar to previous tables. The second line indicates that this is the COST FUNCTION table, that fixed cost (read in on the Y input card) is 0, and that the original price of Y is 20. A later version of F-F II will have an option of a downward sloping demand curve for Y. Thus the original price of Y will be significant. In this output PY is always 20. This is followed by the COEFFICIENTS OF EQUATION (B values), the values of AREAL, L10A, and LEA and finally the PRICE OF RESOURCES.

Then comes the cost values, revenue values and profit for each level of output. The reader will note that average fixed cost (AFC) is not zero even though fixed cost is read in as zero. This is because resource 6 is held constant and thus its cost is added to fixed cost and then the sum divided by output to get average fixed cost. If this were a general expansion path then AFC would show up as 0. Since PY does not change, the marginal revenue (MR) is 20 for all levels of output. Marginal cost (MC) is 19.9742 at an output of 390, and 20.1271 at an output of 400. Thus somewhere between 390 and 400, MC=MR and profit is maximized. The profit level does reach a maximum between these two levels because profit is 1363.44 at the output of 390 and 1362.93 at the output of 400. Therefore between these two output levels,profit reached a peak.

EXPANSION PATH FOR - TEST DATA - CLASS ROOM USAGE FROM DR. BILLINGSLEY
- COST FUNCTIONS * * * FIXED COST IS - 0.000 FIRST PRICE OF Y 20.000

COEFFICIENTS OF EQUATIONS

0.300000000000	C.200000000000	0.100000000000	0.C60000C00000	0.C53500000000
0.035000000000	0.025000000000	0.020000000000	0.C10000000000	0.000000000000

- A - IS 0.66600000000 01 - * - - L10A - IS 0.8235 - * - - LEA - IS 1.8961

PRICES OF INPUTS

9.000	8.000	7.000	6.000	5.500
4.500	3.500	2.500	1.500	0.000

Y	TC	TVC	ATC	AFC	AVC	MC	MR	TR	PROFIT
0.3000000 03	0.4705100 04	0.4255100 04	0.156837D 02	0.1500000 01	0.141837D 02	0.184563D 02	0.2000000 02	C.6000000 04	0.129490D 04
0.3100000 03	0.4890580 04	0.4440580 04	0.1577610 02	0.1451610 01	0.143245D 02	0.186395D 02	0.2000000 02	0.6200000 04	0.130942D 04
0.3200000 03	C.5077880 04	0.4627880 04	0.158684D C2	0.1406250 01	0.144621D 02	0.188186D 02	0.2000000 02	C.6400000 04	0.132212D 04
0.3300000 03	0.5266940 04	0.4816940 04	C.1596040 02	0.1363640 01	0.145968D 02	0.189939D 02	C.2000000 02	0.6600000 04	0.133306D 04
0.3400000 03	0.5457740 04	0.5007740 04	C.1605220 02	0.1323530 01	0.147287D 02	0.191655D 02	0.2000000 02	C.6800000 04	C.134226D 04
C.3500000 03	0.5650240 04	0.5200240 04	0.1614350 02	0.1285710 01	0.148578D 02	0.193335D 02	0.2000000 02	0.7000000 04	0.134976D 04
0.3600000 03	0.5844400 04	0.5394400 04	0.1623450 02	0.1250000 01	0.149845D 02	0.194983D 02	0.2000000 02	0.7200000 04	0.135560D 04
0.3700000 03	0.6040200 04	0.5590200 04	0.1632490 02	0.1216220 01	0.151086D 02	0.196599D 02	0.2000000 02	0.7400000 04	0.135980D 04
0.3800000 03	0.6237590 04	0.5787590 04	0.1641470 02	0.1184210 01	0.152305D 02	0.198185D 02	0.2000000 02	C.7600000 04	0.136241D 04
0.3900000 03	0.6436560 04	0.5986560 04	0.1650400 02	0.1153850 01	0.153501D 02	0.199742D 02	0.2000000 02	0.7800000 04	0.136344D 04
0.4000000 03	0.6637070 04	0.6187070 04	0.1659270 02	0.1125000 01	0.154677D 02	0.201271D 02	C.2000000 02	0.8000000 04	0.136293D 04

(4) VMPX/PX

The following page is an example of the printout of the RATIOS OF THE VALUES OF THE MARGINAL PRODUCTS TO THE PRICE OF EACH RESOURCE and of the MARGINAL PRODUCTS of each resource. The first few lines are similar to previous tables. After the price of inputs comes the labels of the table values. The values come in pairs. The top value is the ratio of the value of the marginal product of a resource to the price of the resource (VMPX/PX). The bottom value is the marginal product (MP) of the resource. The first nine columns are these pairs of values for the first to the ninth resource. The last column contains the output level, Y (on the top) and the marginal revenue, (MR) of the product (on the bottom).

Consider the first pair of values. The VMP1/PX1=1.08364 and the MPX1=0.487638. If the VMP/PX of two resources are equal then the two resources are at the cost minimizing levels. Since X6 is held constant then VMP6/PX6 is not equal to the VMPX/PX of the other resources. The reader will note that the VMPX/PX of each resource (except X6) is above 1 for each output level until 400 is reached. Thus, the profit maximizing level of output is between 390 and 400. Finally, it should be noted that VMPX is obtained by multiplying the MP by MR and since we are assuming pure competition, MR is equal to PY. Therefore MRPX = VMPX.

VMPX/PX AND MP FOR - TEST DATA - CLASS ROOM USAGE FROM DR. BILLINGSLEY
RATIO OF THE VALUE OF THE MARGINAL PRODUCT TO THE PRICE OF EACH RESOURCE

COEFFICIENTS OF EQUATIONS

0.300000000000	0.200000000000	0.100000000000	0.060000000000	0.053500000000
0.035000000000	0.025000000000	0.020000000000	0.010000000000	0.000000000000

- A - IS 0.666000000000 01 - * - - L10A - IS 0.8235 - * - - LEA - IS 1.8961

PRICES OF INPUTS

9.000	8.000	7.000	6.000	5.500
4.500	3.500	2.500	1.500	0.000

VMP1/PX1 MPX1	VMP2/PX2 MPX2	VMP3/PX3 MPX3	VMP4/PX4 MPX4	VMP5/PX5 MPX5	VMP6/PX6 MPX6	VMP7/PX7 MPX7	VMP8/PX8 MPX8	VMP9/PX9 MPX9	Y - OUTPUT MR
0.108364D 01	0.108364D 01	0.108364D 01	0.108364D 01	0.108364D 01	0.466667D 00	0.108364D 01	0.108364D 01	0.108364D 01	0.300000D 03
0.487638D 00	0.433456D 00	0.379274D 00	0.325092D 00	0.298001D 00	0.105000D 00	0.189637D 00	0.135455D 00	0.812730D-01	0.200000D 02
0.107299D 01	0.107299D 01	0.107299D 01	0.107299D 01	0.107299D 01	0.482222D 00	0.107299D 01	0.107299D 01	0.107299D 01	0.310000D 03
0.482845D 00	0.429196D 00	0.375546D 00	0.321897D 00	0.295072D 00	0.108500D 00	0.187773D 00	0.134124D 00	0.804742D-01	0.200000D 02
0.106278D 01	0.106278D 01	0.106278D 01	0.106278D 01	0.106278D 01	0.497778D 00	0.106278D 01	0.106278D 01	0.106278D 01	0.320000D 03
0.478249D 00	0.425111D 00	0.371972D 00	0.318833D 00	0.292264D 00	0.112000D 00	0.185986D 00	0.132847D 00	0.797082D-01	0.200000D 02
0.105297D 01	0.105297D 01	0.105297D 01	0.105297D 01	0.105297D 01	0.513333D 00	0.105297D 01	0.105297D 01	0.105297D 01	0.330000D 03
0.473837D 00	0.421188D 00	0.368540D 00	0.315891D 00	0.289567D 00	0.115500D 00	0.184270D 00	0.131621D 00	0.789728D-01	0.200000D 02
0.104354D 01	0.104354D 01	0.104354D 01	0.104354D 01	0.104354D 01	0.528889D 00	0.104354D 01	0.104354D 01	0.104354D 01	0.340000D 03
0.469595D 00	0.417417D 00	0.365240D 00	0.313063D 00	0.286975D 00	0.119000D 00	0.182620D 00	0.130443D 00	0.782658D-01	0.200000D 02
0.103447D 01	0.103447D 01	0.103447D 01	0.103447D 01	0.103447D 01	0.544444D 00	0.103447D 01	0.103447D 01	0.103447D 01	0.350000D 03
0.465512D 00	0.413788D 00	0.362065D 00	0.310341D 00	0.284480D 00	0.122500D 00	0.181032D 00	0.129309D 00	0.775853D-01	0.200000D 02
0.102573D 01	0.102573D 01	0.102573D 01	0.102573D 01	0.102573D 01	0.560000D 00	0.102573D 01	0.102573D 01	0.102573D 01	0.360000D 03
0.461578D 00	0.410292D 00	0.359005D 00	0.307719D 00	0.282076D 00	0.126000D 00	0.179503D 00	0.128216D 00	0.769297D-01	0.200000D 02
0.101730D 01	0.101730D 01	0.101730D 01	0.101730D 01	0.101730D 01	0.575556D 00	0.101730D 01	0.101730D 01	0.101730D 01	0.370000D 03
0.457784D 00	0.406919D 00	0.356054D 00	0.305190D 00	0.279757D 00	0.129500D 00	0.178027D 00	0.127162D 00	0.762974D-01	0.200000D 02
0.100916D 01	0.100916D 01	0.100916D 01	0.100916D 01	0.100916D 01	0.591111D 00	0.100916D 01	0.100916D 01	0.100916D 01	0.380000D 03
0.454121D 00	0.403664D 00	0.353206D 00	0.302748D 00	0.277519D 00	0.133000D 00	0.176603D 00	0.126145D 00	0.756869D-01	0.200000D 02
0.100129D 01	0.100129D 01	0.100129D 01	0.100129D 01	0.100129D 01	0.606667D 00	0.100129D 01	0.100129D 01	0.100129D 01	0.390000D 03
0.450582D 00	0.400517D 00	0.350453D 00	0.300388D 00	0.275356D 00	0.136500D 00	0.175226D 00	0.125162D 00	0.750976D-01	0.200000D 02
0.993686D 00	0.993686D 00	0.993686D 00	0.993686D 00	0.993686D 00	0.622222D 00	0.993686D 00	0.993686D 00	0.993686D 00	0.400000D 03
0.447159D 00	0.397474D 00	0.347790D 00	0.298106D 00	0.273264D 00	0.140000D 00	0.173895D 00	0.124211D 00	0.745264D-01	0.200000D 02